

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-021130

(43)Date of publication of application : 21.01.2000.

(51)Int.Cl.

G11B 27/00

(21)Application number : 10-239728

(71)Applicant : SONY CORP

(22)Date of filing : 26.08.1998

(72)Inventor : HAMADA TOSHIYA  
FUJINAMI YASUSHI

(30)Priority

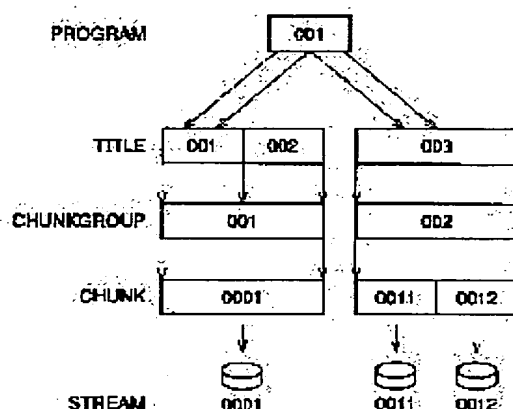
Priority number : 09288180	Priority date : 21.10.1997	Priority country : JP
09288181	21.10.1997	JP
10046857	27.02.1998	JP
10046858	27.02.1998	JP
10120391	30.04.1998	JP
10120389	30.04.1998	JP

(54) INFORMATION PROCESSING APPARATUS AND METHOD, STORAGE MEDIUM AND PROVIDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To simply conduct the editing process within a short period of time through management, by the third management means, of an optional range of the second management means provided for management of the first management means of optional numbers for management of a plurality of data on the one to one basis.

SOLUTION: A user instructs the dividing point in the title dividing process. For example, when the predetermined position of TITLE-002 is designated as the dividing point, an information file of the first title having the starting point of the divided object title as the start point and the dividing point as the ending point is created. Next, an information file (TITLE-003.VDR) of the second title having the dividing point of the division object title (TITLE-002 before division) as the start point, and the ending point thereof as the end point is created. Thereafter, name of the title after the division object title is changed and TITLE-003 is changed to TITLE-004. Thus even if the division process is performed, any change is not effected in CHUNKGROUP, CHUNK, STREAM.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号  
特開2000-21130  
(P2000-21130A)

(43)公開日 平成12年1月21日(2000.1.21)

(51)Int.Cl.<sup>7</sup>

G 1 1 B 27/00

識別記号

F I

G 1 1 B 27/00

フォーマット(参考)

D 5 D 1 1 0

審査請求 未請求 請求項の数 7 O L (全 26 頁)

(21)出願番号 特願平10-239728

(22)出願日 平成10年8月26日(1998.8.26)

(31)優先権主張番号 特願平9-288180

(32)優先日 平成9年10月21日(1997.10.21)

(33)優先権主張国 日本(J P)

(31)優先権主張番号 特願平9-288181

(32)優先日 平成9年10月21日(1997.10.21)

(33)優先権主張国 日本(J P)

(31)優先権主張番号 特願平10-46857

(32)優先日 平成10年2月27日(1998.2.27)

(33)優先権主張国 日本(J P)

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 浜田 俊也

東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

(72)発明者 藤波 靖

東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

(74)代理人 100082131

弁理士 稲本 義雄

Fターム(参考) 5D110 BB20 DA03 DA09 DA15 DB19  
DED4

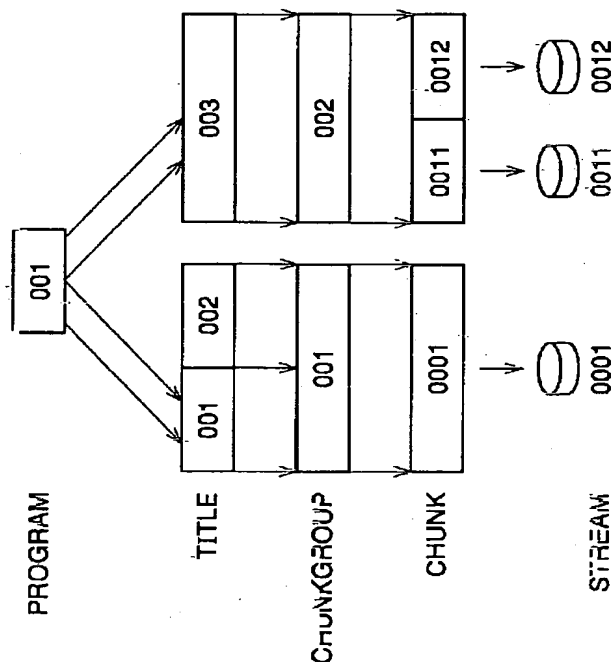
最終頁に続く

(54)【発明の名称】 情報処理装置および方法、記録媒体、並びに提供媒体

(57)【要約】

【課題】 データを簡単に短時間で編集できるようにする。

【解決手段】 実質的なデータで構成されるSTREAMをCHUNKで1対1に管理し、任意の数のCHUNKをCHUNKGROUPで管理する。TITLEには、CHUNKGROUPの所定の範囲を管理させる。PROGRAMには、TITLEの所定の範囲を管理させる。



【特許請求の範囲】

【請求項1】 複数のデータを1対1に管理する第1の管理手段と、

任意の数の前記第1の管理手段を管理する第2の管理手段と、

前記第2の管理手段の任意の範囲を管理する第3の管理手段とを備えることを特徴とする情報処理装置。

【請求項2】 前記第2の管理手段は、任意の数の前記第1の管理手段を、前記第1の管理手段で管理されるデータの少なくとも一部が、時間軸上で重複して再生されるように管理することを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記第1の管理手段は、前記データの単位であり、

前記第3の管理手段は、ユーザからみた前記データの単位であることを特徴とする請求項1に記載の情報処理装置。

【請求項4】 前記第2の管理手段、または、前記第3の管理手段の範囲内において、前記データのシームレス再生が可能とされていることを特徴とする請求項1に記載の情報処理装置。

【請求項5】 複数のデータを1対1に管理する第1の管理ステップと、

前記第1の管理ステップでの任意の数の管理状態を管理する第2の管理ステップと、

前記第2の管理ステップでの任意の範囲の管理状態を管理する第3の管理ステップとを含むことを特徴とする情報処理方法。

【請求項6】 複数のデータを1対1に管理する第1の管理ステップと、

前記第1の管理ステップでの任意の数の管理状態を管理する第2の管理ステップと、

前記第2の管理ステップでの任意の範囲の管理状態を管理する第3の管理ステップとを含む処理を情報処理装置に実行させるプログラムを提供することを特徴とする提供媒体。

【請求項7】 情報と、前記情報を管理する管理情報が記録されている記録媒体において、

前記管理情報は、

複数のデータを1対1に管理する第1の管理手段と、

任意の数の前記第1の管理手段を管理する第2の管理手段と、

前記第2の管理手段の任意の範囲を管理する第3の管理手段とを含むことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置および方法、並びに提供媒体に関し、特に、より簡単に編集を行うことができるようにした情報処理装置および方法、並びに提供媒体に関する。

【0002】

【従来の技術】ビデオデータやオーディオデータを例えばディスクに記録しておくと、それを適宜編集することが可能となる。この編集により、例えば1つのビデオデータの一部の範囲を、他のビデオデータの一部に結合したり、ビデオデータの一部の範囲を削除したりすることができる。

【0003】

【発明が解決しようとする課題】しかしながら、従来、このような編集を行う場合、データそのものを結合したり、あるいは削除するようにしているため、編集（一旦行った編集を再編集する場合を含む）するのに、手間と時間がかかる課題があった。

【0004】本発明はこのような状況に鑑みてなされたものであり、簡単に、かつ、短時間で、編集を行うことができるようにするものである。

【0005】

【課題を解決するための手段】請求項1に記載の情報処理装置は、複数のデータを1対1に管理する第1の管理手段と、任意の数の第1の管理手段を管理する第2の管理手段と、第2の管理手段の任意の範囲を管理する第3の管理手段とを備えることを特徴とする。

【0006】請求項5に記載の情報処理方法は、複数のデータを1対1に管理する第1の管理ステップと、第1の管理ステップでの任意の数の管理状態を管理する第2の管理ステップと、第2の管理ステップでの任意の範囲の管理状態を管理する第3の管理ステップとを含むことを特徴とする。

【0007】請求項6に記載の提供媒体は、複数のデータを1対1に管理する第1の管理ステップと、第1の管理ステップでの任意の数の管理状態を管理する第2の管理ステップと、第2の管理ステップでの任意の範囲の管理状態を管理する第3の管理ステップとを含む処理を情報処理装置に実行させるプログラムを提供することを特徴とする。

【0008】請求項7に記載の記録媒体は、管理情報が、複数のデータを1対1に管理する第1の管理手段と、任意の数の第1の管理手段を管理する第2の管理手段と、第2の管理手段の任意の範囲を管理する第3の管理手段とを含むことを特徴とする。

【0009】請求項1に記載の情報処理装置においては、複数のデータを1対1に管理する第1の管理手段を管理する第2の管理手段の任意の範囲が、第3の管理手段で管理される。

【0010】請求項5に記載の情報処理方法、および請求項6に記載の提供媒体においては、第1の管理ステップでの任意の数の管理状態が、第2の管理ステップで管理され、第2の管理ステップでの任意の範囲の管理状態が、第3の管理ステップで管理される。

【0011】請求項7に記載の記録媒体においては、情

報を管理する管理情報として、第1の管理手段、第2の管理手段、および第3の管理手段が記録されている。

#### 【0012】

【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0013】請求項1に記載の情報処理装置は、複数のデータを1対1に管理する第1の管理手段（例えば、図24のCHUNK\_%%%.ABST）と、任意の数の第1の管理手段を管理する第2の管理手段（例えば、図21のCHUNKGROUP\_###.CGIT）と、第2の管理手段の任意の範囲を管理する第3の管理手段（例えば、図15のTITLE\_###.VDR）とを備えることを特徴とする。

【0014】最初に本発明において情報が記録または再生される記録媒体（メディア）上のファイル配置について説明する。メディア上には、図1に示すように、次の7種類のファイルが記録される。

VOLUME.TOC

ALBUM.STR

PROGRAM\_\$\$\$ .PGI

TITLE\_###.VDR

CHUNKGROUP\_@@@.CGIT

CHUNK\_%%%.ABST

CHUNK\_%%%.MPEG2

【0015】ルートディレクトリにはVOLUME.TOCおよびALBUM.STRが置かれる。また、ルートディレクトリ直下のディレクトリ"PROGRAM"には、"PROGRAM\_\$\$\$ .PGI"（ここで"\$\$\$"はプログラム番号を表す）が置かれる。同様に、ルートディレクトリ直下のディレクトリ"TITLE"には、"TITLE\_###.VDR"（ここで"###"はタイトル番号を表す）が、ディレクトリ"CHUNKGROUP"には、"CHUNKGROUP\_@@@.CGIT"（ここで"@@@"はチャンクグループ番号を表す）が、ディレクトリ"CHUNK"には、"CHUNK\_%%%.ABST"（ここで"%%%"はチャンク番号を表す）が、それぞれ置かれる。

【0016】ルートディレクトリ直下のMPEGAVディレクトリには、更に1つ以上のサブディレクトリが作成され、その下に、"CHUNK\_%%%.MPEG2"（ここで"%%%"はチャンク番号を表す）が置かれる。

【0017】VOLUME.TOCのファイルは、メディア上に1つ有るのが普通である。ただし、ROMとRAMのハイブリッド構造のメディア等、特殊な構造のメディアでは、複数存在することも有り得る。このファイルは、メディアの全体の性質を示すために用いられる。

【0018】VOLUME.TOCの構造は図2に示すようになっている。先頭にfile\_type\_idが置かれ、これにより該当

ファイルがVOLUME.TOCであることが示される。次にvolume\_information()が続き、最後にtext\_block()が続く。

【0019】図3にvolume\_information()の構成が示されている。これは、volume\_attribute()、resume()、volume\_rating()、write\_protect()、play\_protect()、recording\_timer()を含んでいる。

【0020】volume\_attribute()は、logical volumeの属性を記録する領域であり、図4にその詳細な構造が示されている。同図に示すように、この領域には、title\_playback\_mode\_flag、program\_playback\_mode\_flagなどが含まれている。

【0021】resume()は、メディアの再挿入時に、eject直前の状態を復元するための情報を記録する領域であり、その詳細な構造は、図5に示されている。

【0022】図3のvolume\_rating()は、volume全体に対する視聴年齢制限を年齢やカテゴリに応じて実現するための情報を記録する領域であり、その詳細な構造は、図6に示されている。

【0023】図3のwrite\_protect()は、volume内に記録されているtitle、programに対する変更や、消去操作を制限する情報を記録する領域であり、その詳細な構造は、図7に示されている。

【0024】図3のplay\_protect()は、volume内に記録されているtitle、programに対する再生許可、不許可の設定、あるいは、再生回数を制限する情報を記録する領域であり、その詳細な構造は、図8に示されている。

【0025】図3のrecording\_timer()は、記録時間を制御する情報を記録する領域であり、その詳細な構造は、図9に示されている。

【0026】図2のVOLUME.TOCのtext\_block()の詳細な構造は図10に示されている。このtext\_block()には、language\_set()とtext\_itemが含まれており、その詳細な構造は図11と図12にそれぞれ示されている。

【0027】図1のALBUM.STRのファイルは、メディア上に1つ有るのが普通である。ただし、ROMとRAMのハイブリッド構造のメディア等、特殊な構造のメディアでは、複数存在することも有り得る。このファイルは、複数のメディアを組み合わせ、あたかも1つのメディアであるような構成にするために使用される。

【0028】このALBUM.STRの構造は、図13に示すようになっている。先頭にfile\_type\_idが置かれ、該当ファイルがALBUM.STRであることを示す。次にalbum()が続き、最後にtext\_block()が続く。

【0029】album()は、複数のvolume（複数のメディア）を1つのまとまりとして扱うための情報を記録する領域であり、その詳細な構造は、図14に示されている。

【0030】図1のTITLE\_###.VDRのファイルは、タイトルの数だけ存在する。タイトルとは、例えばcompact discで言うところの1曲や、テレビ放送の1番組を言

う。この情報の構造は図15に示すようになっている。先頭にfile\_type\_idが置かれ、これにより該当ファイルがTITLE\_###.VDRであることが示される。次にtitle\_info()が続く、最後にtext\_block()が続く。###はタイトル番号を示す文字列である。

【0031】title\_info()は、chunkgroup上における、titleの開始点、終了点、その他titleに関する属性を記録するための領域であり、その詳細な構造は、図16に示されている。

【0032】図1のPROGRAM\_\$\$\$ .PGIのファイルは、プログラムの数だけ存在する。プログラムは、タイトルの一部（あるいは全部）の領域を指定した複数のカットで構成され、各カットは指定された順番で再生される。この情報の構造は図17に示されている。先頭にfile\_type\_idが置かれ、該当ファイルがPROGRAM\_\$\$\$ .PGIであることを示す。次にprogram()が続く、最後にtext\_block()が続く。\$\$\$はタイトル番号を示す文字列である。

【0033】program()は、素材に対して不可逆な編集を施すことなしに、titleの必要な部分を集めて再生するのに必要な情報を記録する領域であり、その詳細な構造は、図18に示されている。

【0034】図18のprogram()は、1つのplay\_listを有している。このplay\_list()の詳細は、図19に示されている。

【0035】play\_listには、play\_item()が複数置かれている。play\_item()の詳細は、図20に示されている。

【0036】図1のCHUNKGROUP\_@@@ .CGITのファイルは、チャンクグループの数だけ存在する。チャンクグループはビットストリームを並べるためのデータ構造である。このファイルは、ユーザがVDR（ビデオディスクレコーダ）など、メディアを記録再生する装置を普通に操作している分にはユーザに認識されない。

【0037】この情報の構造は図21に示すようになっている。先頭にfile\_type\_idが置かれ、該当ファイルがCHUNKGROUP\_@@@ .CGITであることを示す。その次にchunkgroup\_time\_base\_flagsとchunkgroup\_time\_base\_offsetが有り、次にchunk\_connection\_info()、最後にtext\_block()が続く。

【0038】chunkgroup\_time\_base\_flagsは、chunkgroupの基準カウンタに関するflagを示し、chunkgroup\_time\_base\_offsetは、chunkgroup内の基準時間軸の開始時刻を示す。これは、90kHzでカウントアップするカウンタにセットする値であり、32ビットの大きさを有する。chunk\_connection\_info()は、videoの切替点や、videoとaudioの同期など、特異な点の情報を記憶する領域であり、その詳細な構造は、図22に示されている。

【0039】このchunk\_connection\_info()には、チャンクグループに属するチャンクの数だけchunk\_arrangement\_info()のループが置かれる。図23にこのchunk\_ar-

rangement\_info()の詳細が示されている。

【0040】図1のCHUNK\_%%% .ABSTのファイルは、チャンクの数だけ存在する。チャンクはストリームファイル1つに対応する情報ファイルである。この情報の構造は図24に示すようになっている。先頭にfile\_type\_idが置かれ、これにより、該当ファイルがCHUNK\_%%% .ABSTであることが示される。

【0041】図1のCHUNK\_%%% .MPEG2のファイルは、ストリームファイルである。このファイルはMPEGのビットストリームを格納しており、この他のファイルが情報のみを記録しているのとは異なっている。

【0042】図25は、以上のようなファイルを有するメディアとしての光ディスクに対して情報を記録または再生する光ディスク装置の構成例を表している。この光ディスク装置では、1枚の書き換え型の光ディスク1に対して1系統の光ヘッド2が設けられており、データの読み出しと書き込みの双方にこの光ヘッド2が共用される。

【0043】光ヘッド2により光ディスク1から読み出されたビットストリームは、RFおよび復調／変調回路3で復調された後、ECC回路4で誤り訂正が施され、スイッチ5を介して、読み出しレートとデコード処理レートとの差を吸収するための読み出しチャンネル用バッファ6に送られる。読み出しチャンネル用バッファ6の出力はデコード7に供給されている。読み出しチャンネル用バッファ6はシステムコントローラ13から読み書きができるように構成されている。

【0044】読み出しチャンネル用バッファ6から出力されたビットストリームは、デコーダ7でデコードされ、そこからビデオ信号とオーディオ信号が出力される。デコーダ7から出力されたビデオ信号は合成回路8に入力され、OSD (On Screen Display) 制御回路9が出力するビデオ信号と合成された後、出力端子P1から図示せぬディスプレイに出力され、表示される。デコーダ7から出力されたオーディオ信号は、出力端子P2から図示せぬスピーカに送られて再生される。

【0045】他方、入力端子P3から入力されたビデオ信号、および入力端子P4から入力されたオーディオ信号は、エンコーダ10でエンコードされた後、エンコード処理レートと書き込みレートとの差を吸収するための書き込みチャンネル用バッファ11に送られる。この書き込みチャンネル用バッファ11もシステムコントローラ13から読み書きができるように構成されている。

【0046】書き込みチャンネル用バッファ11に蓄積されたデータは、書き込みチャンネル用バッファ11から読み出され、スイッチ5を介してECC回路4に入力されて誤り訂正符号が付加された後、RFおよび復調／変調回路3で変調される。RFおよび復調／変調回路3より出力された信号（RF信号）は、光ヘッド2により光ディスク1に書き込まれる。

【0047】アドレス検出回路12は、光ディスク1の記録または再生するトラックのアドレス情報を検出する。システムコントローラ13は、この光ディスク装置の各部の動作を制御するものであり、各種の制御を行うCPU21、CPU21が実行すべき処理プログラム等を格納したROM22、処理過程で生じたデータ等を一時記憶するためのRAM23、および光ディスク1に対して記録または再生する各種の情報ファイルを記憶するRAM24を有している。CPU21は、アドレス検出回路12の検出結果に基づいて、光ヘッド2の位置を微調整する。CPU21はまた、スイッチ5の切り替え制御を行う。各種のスイッチ、ボタンなどから構成される入力部14は、各種の指令を入力するとき、ユーザにより操作される。

【0048】次に、基本的な情報ファイルの読み込み動作について説明する。例えば、“VOLUME.TOC”情報ファイルの読み込みを行うとき、システムコントローラ13のCPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、“VOLUME.TOC”が記録されている光ディスク1上の物理アドレスと、その長さを確定する。続いて、CPU21は、この“VOLUME.TOC”のアドレス情報に基づき、光ヘッド2を読み出し位置に移動させる。そしてCPU21は、光ヘッド2、RFおよび復調/変調回路3、並びにECC回路4を読み出しモードに設定するとともに、スイッチ5を読み出しチャンネル用バッファ6側に切り替え、さらに光ヘッド2の位置を微調整した後、光ヘッド2による読み出しを開始させる。これにより“VOLUME.TOC”の内容が光ヘッド2により読み出され、RFおよび復調/変調回路3により復調され、さらにECC回路4により誤り訂正が行われた後、読み出しチャンネル用バッファ6に蓄積される。

【0049】読み出しチャンネル用バッファ6に蓄積されたデータ量が、“VOLUME.TOC”の大きさと等しいか、あるいはより大きくなった時点で、CPU21は読み出しを停止させる。その後、CPU21は、読み出しチャンネル用バッファ6から該当データを読み出し、RAM24に記憶させる。

【0050】次に、基本的な情報ファイル書き込み動作について、“VOLUME.TOC”情報ファイルを書き込む場合を例として説明する。CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク1）中に、これから書こうとしている“VOLUME.TOC”と等しいか、より大きい大きさを持つ空き領域を探し、そのアドレスを確定する。

【0051】次に、CPU21は、RAM24に用意されている、新たに書き込むべき“VOLUME.TOC”を、書き込みチャンネル用バッファ11に転送する。続いて、CPU21は、空き領域のアドレス情報に基づき、光ヘッド2を書き込み位置に移動させる。そしてCPU21は、光ヘッド2、RFおよび復調/変調回路3、並びにECC回路4を書き込みモードに設定するとともに、スイッチ5を書き込み

チャンネル用バッファ11側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による書き込みを開始させる。

【0052】これにより新たに用意した“VOLUME.TOC”の内容が、書き込みチャンネル用バッファ11から読み出され、スイッチ5を介してECC回路4に入力され、誤り訂正符号が付加された後、RFおよび復調/変調回路3により変調される。RFおよび復調/変調回路3より出力された信号は、光ヘッド2により光ディスク1に記録される。書き込みチャンネル用バッファ11から読み出され、光ディスク1に記録されたデータ量が、“VOLUME.TOC”の大きさと等しくなった時点で、CPU21は書き込み動作を停止させる。

【0053】最後に、CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク1）中の“VOLUME.TOC”を指し示すポインタを、新しく書込んだ位置を指し示すように書き換える。

【0054】次に、基本的なストリーム再生動作について、図1のCHUNK\_0001.MPEG2というストリームを再生する場合を例として説明する。CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、“CHUNK\_0001.MPEG2”が記録されている光ディスク1上の物理アドレスと、その長さを確定する。続いて、CPU21は、この“CHUNK\_0001.MPEG2”のアドレス情報に基づき、光ヘッド2を読み出し位置に移動させる。そして光ヘッド2、RFおよび復調/変調回路3、並びにECC回路4を読み出しモードに設定するとともに、スイッチ5を読み出しチャンネル用バッファ6側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による読み出しを開始させる。

【0055】光ヘッド2により読み出された“CHUNK\_0001.MPEG2”の内容が、RFおよび復調/変調回路3、ECC回路4、並びにスイッチ5を介して読み出しチャンネル用バッファ6に蓄積される。読み出しチャンネル用バッファ6に蓄積されたデータは、デコーダ7に出力され、デコード処理が施されて、ビデオ信号とオーディオ信号がそれぞれ出力される。オーディオ信号は出力端子P2から出力され、ビデオ信号は、合成回路8を介して出力端子P1から出力される。

【0056】光ディスク1から読みだされ、デコード、表示されたデータ量が、“CHUNK\_0001.MPEG2”の大きさと等しくなった時点で、あるいは、入力部14から読み出し動作の停止が指定された時点で、CPU21は、読み出しおよびデコード処理を停止させる。

【0057】次に、基本的なストリーム記録動作を、“CHUNK\_0001.MPEG2”情報ファイルを書き込む場合を例として説明する。CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク1）中にこれから書こうとし

ている"CHUNK\_0001.MPEG2"と等しいか、それより大きい大きさを持つ空き領域を探し、そのアドレスを確定する。

【0058】入力端子P3から入力されたビデオ信号、および入力端子P4から入力されたオーディオ信号は、エンコーダ10によりエンコードされた後、書き込みチャンネル用バッファ11に蓄積される。続いて、CPU21は、空き領域のアドレス情報に基づき、光ヘッド2を書き込み位置に移動させる。そしてCPU21は、光ヘッド2、RFおよび復調／変調回路3、並びにECC回路4を書き込みモードに設定するとともに、スイッチ5を書き込みチャンネル用バッファ11側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による書き込みを開始させる。これにより新たに用意した"CHUNK\_0001.MPEG2"の内容が、書き込みチャンネル用バッファ11から読み出され、スイッチ5、ECC回路4、RFおよび復調／変調回路3を介して光ヘッド2に入力され、光ディスク1に記録される。

【0059】書き込みチャンネル用バッファ11から読み出され、光ディスク1に記録されたデータ量が、予め設定した値と等しくなったとき、あるいは入力部14から書き込み動作の停止が指定されたとき、CPU21は書き込み動作を停止させる。最後に、CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク1）中の"CHUNK\_0001.MPEG2"を指し示すポインタを、新しく書込んだ位置を指し示すように書き換える。

【0060】いま、光ディスク1に、図26に示すような情報ファイルとストリームファイルが記録されているものとする。この例では、"PROGRAM\_001.PGI"という名前の1つのプログラムのファイルが含まれている。また、この光ディスク1には、"TITLE\_001.VDR"、"TITLE\_002.VDR"、および"TITLE\_003.VDR"という名前の3つのタイトルのファイルが含まれている。

【0061】さらに、この光ディスク1には、"CHUNKGROUP\_001.CGIT"と"CHUNKGROUP\_002.CGIT"という2つのチャンクグループのファイルが含まれている。また、この光ディスク1には、"CHUNK\_0001.MPEG2"、"CHUNK\_0011.MPEG2"、および"CHUNK\_0012.MPEG2"という名前の3つのストリームのファイルが含まれているとともに、それぞれに対応する情報として、"CHUNK\_0001.ABST"、"CHUNK\_0011.ABST"、および"CHUNK\_0012.ABST"の3つの情報ファイルが置かれている。

【0062】図26に示した情報ファイルとストリームファイルを有する光ディスク1の論理構造は、図27に示すようになる。この例では、チャンク情報ファイル"CHUNK\_0001.ABST"は、ストリームファイル"CHUNK\_0001.MPEG2"を、またチャンク情報ファイル"CHUNK\_0011.ABST"は、ストリームファイル"CHUNK\_0011.MPEG2"を、さらに、チャンク情報ファイル"CHUNK\_0012.ABST"は、スト

リームファイル"CHUNK\_0012.MPEG2"を、それぞれ指定している。具体的には、図24のCHUNK\_%%.ABST中の、chunk\_file\_idというフィールドで、ストリームのファイルIDが指定される。

【0063】さらに、この例では、チャンクグループ情報ファイル"CHUNKGROUP\_001.CGIT"は、チャンク情報ファイル"CHUNK\_0001.ABST"を、またチャンクグループ情報ファイル"CHUNKGROUP\_002.CGIT"は、チャンク情報ファイル"CHUNK\_0011.ABST"と"CHUNK\_0012.ABST"を、それぞれ指定している。具体的には、図23のchunk\_arrangement\_info()の中のchunk\_info\_file\_idというフィールドでチャンク情報のファイルIDが指定される。このchunk\_arrangement\_info()はチャンクグループ情報ファイルの中にあり、該当チャンクグループに属するチャンクの数だけ存在するデータ構造となっている（図23のchunk\_arrangement\_info()は、図22のchunk\_connection\_info()に記述されており、このchunk\_connection\_info()は、図21のCHUNKGROUP\_###.CGITに記述されている）。

【0064】CHUNKGROUP\_001には、chunk\_arrangement\_info()が1つだけあり、その中のchunk\_info\_file\_idがCHUNK\_0001を指定している。CHUNKGROUP\_002には、chunk\_arrangement\_info()が2つあり、その中で、それぞれCHUNK\_0011とCHUNK\_0012が指定されている。このような場合のため、チャンクグループは、複数のチャンクの再生順序等を指定できるようになっている。

【0065】具体的には、まず、図21のCHUNKGROUP\_###.CGIT中のchunkgroup\_time\_base\_offsetにより、該当チャンクグループでの時計の初期値が定められる。次に、各チャンクを登録する際に、図23のchunk\_arrangement\_info()のpresentation\_start\_cg\_countとpresentation\_end\_cg\_time\_countが指定される。

【0066】例えば、図28に示すように、CHUNK\_0011の長さ（時間）をA、CHUNK\_0012の長さ（時間）をBとする。CHUNK\_0011のpresentation\_start\_cg\_countがchunkgroup\_time\_base\_offsetに等しく、presentation\_end\_cg\_countが"chunkgroup\_time\_base\_offset+A"に等しい。またCHUNK\_0012のpresentation\_start\_cg\_countがchunkgroup\_time\_base\_offset+Aに等しく、presentation\_end\_cg\_countが"chunkgroup\_time\_base\_offset+A+B"に等しい。このように設定すると、CHUNKGROUP\_002は、CHUNK\_0011とCHUNK\_0012を連続的に再生させたものとして定義される。

【0067】なお、CHUNK\_0011とCHUNK\_0012の再生時刻に重なりがある場合には、時刻をそのようにずらすことで指定ができる。また、図23のchunk\_arrangement\_info()中のtransition\_info()に記述を行うことで、2つのストリーム間の遷移において、特殊効果（フェードイン、フェードアウト、ワイプ等）を指定できるようになっている。

【0068】図26 (図27) の例では、タイトル情報ファイル"TITLE\_001.VDR"と"TITLE\_002.VDR"は、チャンクグループ情報ファイル"CHUNKGROUP\_001.CGIT"を、またタイトル情報ファイル"TITLE\_003.VDR"はチャンクグループ情報ファイル"CHUNKGROUP\_002.CGIT"を、それぞれ指定している。具体的には、図16のtitle\_info()中において、cgit\_file\_idというフィールドで、チャンクグループのファイルIDを指定し、さらにtitle\_start\_chunk\_group\_time\_stampとtitle\_end\_chunk\_group\_time\_stampというフィールドで、チャンクグループ内で該当タイトルが定義される時間的な範囲を指定している。

【0069】例えば、図27の例では、CHUNKGROUP\_001の前半をTITLE\_001が、後半をTITLE\_002が、それぞれ指し示している。なお、この分割はユーザからの要求により行われたものであり、その位置はユーザにとって任意であり、予め決めておくことはできない。ここでTITLE\_001とTITLE\_002による分割の位置を、CHUNKGROUP\_001の先頭からAだけ離れた位置に設定したとする。

【0070】TITLE\_001はチャンクグループとしてCHUNKGROUP\_001を指定し、タイトルの開始時刻として、CHUNKGROUP\_001の開始時刻を指定し、タイトルの終了時刻として、ユーザから指定された点の時刻を指定する。

【0071】つまりTITLE\_001のtitle\_start\_chunk\_group\_time\_stampとして、CHUNKGROUP\_001のchunkgroup\_time\_base\_offset (先頭の位置) が設定され、TITLE\_001のtitle\_end\_chunk\_group\_time\_stampとして、CHUNKGROUP\_001のchunkgroup\_time\_base\_offsetにAの長さを加えたものが設定される。

【0072】また、TITLE\_002はチャンクグループとしてCHUNKGROUP\_001を指定し、タイトルの開始時刻として、ユーザから指定された点の時刻を指定し、タイトルの終了時刻として、CHUNKGROUP\_001の終了時刻を指定する。

【0073】つまりTITLE\_002のtitle\_start\_chunk\_group\_time\_stampとして、CHUNKGROUP\_001のchunkgroup\_time\_base\_offset (先頭の位置) にAの長さを加えたものが設定され、TITLE\_002のtitle\_end\_chunk\_group\_time\_stampとして、CHUNKGROUP\_001のchunkgroup\_time\_base\_offsetにCHUNKGROUP\_001の長さを加えたものが設定される。

【0074】さらに、TITLE\_003はチャンクグループとしてCHUNKGROUP\_002を指定し、タイトルの開始時刻としてCHUNKGROUP\_002の開始時刻を指定し、タイトルの終了時刻としてCHUNKGROUP\_002の終了時刻を指定する。

【0075】つまりTITLE\_003のtitle\_start\_chunk\_group\_time\_stampとして、CHUNKGROUP\_002のchunkgroup\_time\_base\_offsetが設定され、TITLE\_003のtitle\_end\_chunk\_group\_time\_stampとして、CHUNKGROUP\_002のchunkgroup\_time\_base\_offsetにCHUNKGROUP\_002の長さを加えたものが設定される。

【0076】さらに、この例では、プログラム情報ファイル"PROGRAM\_001.PGI"は、TITLE\_001の一部とTITLE\_003の一部を、この順番で再生するように指定している。具体的には、図20のplay\_item()中のtitle\_numberによりタイトルを指定し、各タイトルで定義される時刻で開始点と終了点を定義することで、1つのカットが抜き出される。このようなカットを複数個集めて、プログラムが構成される。

【0077】次に、光ディスク1に、新たな情報を追記録 (アペンド記録) する場合の動作について説明する。この記録は、具体的には、例えば、タイマ録画により、あるいはユーザが入力部14を操作して、光ディスク装置に対してリアルタイムに録画を指令することにより行われる。後者の場合、録画ボタンが押されたようなときは、録画終了時刻を予測することはできないが、ワンタッチ録画機能 (操作後、一定時間だけ録画が行われる機能) のボタンが押されたときは、終了時刻を予測することができる。

【0078】ここではタイマ録画を例にとって説明する。この場合、光ディスク装置のユーザは事前に、録画開始時刻、録画終了時刻、ビットストリームのビットレート、録画を行うチャネル等を指定してあるものとする。また、録画の予約を行った時点で、ビットレートと録画時間に見合う空き容量が光ディスク1に残されていることが、予め確認されているものとする。

【0079】記録予約時と予約された記録の実行時との間に、光ディスク1に対して更なる記録が行われたような場合、今回予約された番組を、指定されたビットレートで記録する分の容量を確保することができなくなる場合がある。このような場合、CPU21は、ビットレートを、指定された値より下げて、予約された時間分の情報を記録するようにするか、または、ビットレートはそのままにして、記録可能な時間だけ記録するようにする。CPU21は、このとき、更なる記録が行われ、予約した記録に不具合が出た時点でユーザにその旨を伝えるメッセージを発することは言うまでもない。

【0080】さて、予約された録画の開始時刻が近づくと、CPU21は内蔵するタイマやクロックを使用して、モードを、スリープモードから動作モードに自動的に復帰させる。そしてCPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、予約された番組が記録できるだけの領域を光ディスク1上に確保する。つまり、予約録画の終了時刻から開始時刻を減算した結果 (録画時間) にビットレートを乗じた数値が、予約された番組を記録するのに必要な領域の大きさであり、CPU21はこの大きさの領域をまず確保する。その他、この記録に際して、ストリームファイル以外に情報ファイルを記録する必要がある場合、例えば新たなタイトルとして登録するためにタイトル情報ファイル等が必要である場合には、それらの情報ファイルが記

録できるだけの容量を光ディスク1に確保しておく必要がある。必要な分の領域を確保することができない場合には、上述したような方法（ビットレートの変更、録画可能な時間内だけの録画などの方法）で対応が取られることになる。

【0081】なおこのとき、新しいタイトルの記録なので、ユーザは、新たなストリームディレクトリの新たなストリームファイルとして新しいストリームファイルのファイル名を付ける。ここでは、これを、¥MPEGAV¥STREAMS\_003¥CHUNK\_0031とする。つまり、図29に示すように、ルートディレクトリの下にMPEGAVディレクトリの下にSTREAM\_003ディレクトリの下にCHUNK\_0031.MPEG2という名前のファイルとする。

【0082】CPU21は、各部に対して記録モードの実行を命令する。例えば、図示せぬチューナから入力端子P3に入力されたビデオ信号、および入力端子P4に入力されたオーディオ信号は、エンコーダ10によりエンコードされた後、書き込みチャンネル用バッファ11に蓄積される。続いて、CPU21は、先程確保した領域のアドレス情報に基づき、光ヘッド2を書き込み位置に移動させる。そしてCPU21は、光ヘッド2、RFおよび復調／変調回路3、並びにECC回路4を書き込みモードに設定するとともに、スイッチ5を書き込みチャンネル用バッファ11側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による書き込みを開始させる。これにより新たに用意した“CHUNK\_0031.MPEG2”の内容が、書き込みチャンネル用バッファ11から読み出され、スイッチ5、ECC回路4、RFおよび復調／変調回路3、並びに光ヘッド2を介して、光ディスク1に記録される。

【0083】以上の書き込み動作を続けて、以下のいずれかの条件が発生した時点で、CPU21は、書き込み動作を停止させる。

- 1) 予約された記録の終了時刻になったとき
- 2) 容量不足、その他の原因により光ディスク1に記録ができなくなったとき
- 3) 録画動作の停止が指令されたとき

【0084】次に、CPU21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム中の“CHUNK\_0031.MPEG2”を指し示すポインタを新しく書込んだ位置を指し示す値に書き換える。また、CPU21は、チャンク情報、チャンクグループ情報、タイトル情報のそれぞれのファイルを用意し、しかるべき名前をつけて記録する。なお、記録時あるいは予約時に、光ディスク1上に、これらのファイルを記録することができるだけの空き容量を確保しておく必要がある。

【0085】このようにして、例えば図30に示すように、新たな情報ファイルが作成される。同図において、ファイル名の右肩にアスタリスク(\*)をつけたものが、今回新たに作成されたファイルである。

【0086】図31は、新たにでき上がった情報ファイルの関係を示したものである。TITLE\_004はCHUNKGROUP\_003を指定し、CHUNKGROUP\_003はCHUNK\_0031を指定し、CHUNK\_0031はSTREAM\_0031を指定している。

【0087】すなわち、新たなストリームはTITLE\_004として、情報ファイルに登録されている。ユーザは光ディスク装置のタイトルを確認する機能により、TITLE\_004の属性等を知ることができ、また、TITLE\_004を再生することができる。

【0088】次に、図26（図27）に例示するような光ディスク1上に、上書き記録する場合の動作について説明する。上書き記録とは、ビデオテープに信号を記録する場合と同様に、それまでに記録されている番組の上に（その番組を消去して）新たな番組を記録していく動作のことを言う。

【0089】上書き記録では、上書き記録を開始する位置が重要である。例えばユーザからTITLE\_001の先頭から上書き記録を開始することが指定されたとする。この時上書き記録は、TITLE\_001、TITLE\_002、TITLE\_003をそれぞれ順に書き換えながら行われる。TITLE\_003の最後まで書き換えてもまだ記録動作が終了しない場合には、光ディスク1上の空き領域に新たな領域を確保して記録が続行される。例えばTITLE\_002が記録開始位置とされた場合には、TITLE\_001は記録開始位置より前に位置するので、今回の記録動作により書き換えられることはない。

【0090】いま、TITLE\_003の先頭からタイマ録画により上書きするものとする。この場合、光ディスク装置のユーザは事前に、録画開始時刻、終了時刻、ビットストリームのビットレート、録画を行うチャンネル等を指定しているものとする。また、上書き記録では重要な記録開始位置がTITLE\_003の先頭と指定されたものとする。さらにこの場合においても、録画の予約を行った時点で、ビットレートと録画時間に見合う容量が光ディスク1上に存在することが、予め確認されているものとする。上書き記録の場合には、指定された位置から上書き可能な（複数の）タイトルの総容量と、光ディスク1の空き容量の和が記録可能容量となる。つまり、今回の場合には、TITLE\_003が管理するストリームSTREAM\_0011とSTREAM\_0012の総容量と、光ディスク1上の空き容量の和が記録可能な容量となる。

【0091】上書き記録では、記録可能な容量分に対して、どのような順番で実際の記録を行なっていくかという選択肢がいくつかある。まず、最初に考えられるのがタイトルで指定されているストリームの順番に記録していく方法である。つまり、今回の場合には、まずSTREAM\_0011の先頭から記録を開始し、STREAM\_0011の終わりまで記録したら、STREAM\_0012の先頭から記録を続行し、STREAM\_0012の終わりまで記録したら、今度は空き領域に記録を行なう方法である。もう1つの方法は、まず、空

き領域に記録を行い、空き領域が無くなった時点で、現存するストリーム上に記録していく方法である。

【0092】前者の方法は、ビデオテープのエミュレーションという意味で優れている。つまり、ビデオテープと同様の動作であるという意味で、ユーザから理解され易いという特徴を有する。後者の方法は、既に記録されているストリームの消去が後回しにされるため、記録されているものの保護という点で優れていると言う特徴を有する。

【0093】なお、記録予約時と予約された記録の実行時との間に、光ディスク1に対して更なる記録が行われた場合に、今回予約された番組を、指定されたビットレートで記録する分の容量を確保することができない場合がある。このような場合、上述した場合と同様に、予約実行時に、ビットレートが自動的に下げられ、予約された時間分だけすべて記録されるか、または、ビットレートはそのままにして、記録可能な時間だけ記録が行われる。

【0094】予約された録画の開始時刻が近づくと、光ディスク装置はスリープモードから動作モードに復帰する。CPU21は、光ディスク1上の空き容量をすべて確保する。もちろん、この時点で空き容量を確保せず、必要になった時点で確保するという方法もあるが、ここでは説明のために、記録開始以前に必要な領域を確保するものとする。

【0095】なお、タイマ録画等で、開始時刻、終了時刻、ビットレートが指定されているため、必要な領域の大きさが予め判っている場合には、必要な分だけ（あるいは幾分かのマージンを加えた分だけ）容量を確保するようにしてもよい。この記録に際して情報ファイルを記録する必要がある場合、例えば新たなタイトルとして登録するためにタイトル情報ファイル等が必要である場合、それらの情報ファイルも記録することができるだけの容量を残しておく必要がある。

【0096】ここでは、新たなストリームディレクトリの新たなストリームファイルとして新しいストリームファイルのファイル名をつけるものとする。つまり、ここでは、ファイル名を、¥MPEGAV¥STREMS\_002¥CHUNK\_0031とする。すなわち、図32に示すように、ルートディレクトリの下にMPEGAVディレクトリの下にSTREAM\_002ディレクトリの下にCHUNK\_0031.MPEG2という名前のファイルが作成される。

【0097】入力端子P3に入力されたビデオ信号、および入力端子P4に入力されたオーディオ信号は、エンコーダ10によりエンコードされた後、書き込みチャネル用バッファ11に蓄積される。続いて、CPU21は、先程確保した領域のアドレス情報に基づき、光ヘッド2を書き込み位置に移動させる。そしてCPU21は、光ヘッド2、RFおよび復調／変調回路3、並びにECC回路4を書き込みモードに設定するとともに、スイッチ5を

書き込みチャネル用バッファ11側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による書き込みを開始させる。これにより新たに用意した"CHUNK\_0031.MPEG2"の内容が、書き込みチャネル用バッファ11から読み出され、スイッチ5、ECC回路4、RFおよび復調／変調回路3、並びに光ヘッド2を介して、光ディスク1に記録される。

【0098】この時、まずはストリームファイル"CHUNK\_0011.MPEG2"が書き換えられる。そして"CHUNK\_0011.MPEG2"の最後まで記録が行われたら、次に、"CHUNK\_0012.MPEG2"へ記録が進められ、さらに、"CHUNK\_0031.MPEG2"へと記録が進められる。

【0099】以上の動作を続けて、上述した場合と同様に、3つの条件のいずれかが発生した時点で、CPU21は、書き込み動作を停止させる。

【0100】次に、CPU21は、予めその処理プログラムに組み込んであったファイルシステム操作命令を使用し、ストリームファイル、チャンク情報、チャンクグループ情報、タイトル情報を更新する。

【0101】ところで、書き込みが終了したタイミングによって、ファイルの構成が変化する。例えば、CHUNK\_0011.MPEG2とCHUNK\_0012.MPEG2の2つのストリームの上書きを終了した後、さらにCHUNK\_0031.MPEG2に記録が行われた場合、光ディスク1のファイルの構成は、図33に示すようになる。ファイル名の右肩にアスタリスク(\*)をつけたものが今回新たに作成されたファイルである。

【0102】図34は、このようにして新たにでき上がったファイル（図33のファイル）の関係を示したものである。図31と比較して明らかなように、TITLE\_003が指定しているCHUNKGROUP\_002に含まれるCHUNKとしてCHUNK\_0031が増えており、CHUNK\_0031はSTREAM\_0031を指定している。

【0103】一方、既存ストリームの上書きの途中で上書き記録が終了した場合、例えば、CHUNK\_0011の記録の途中で上書き記録が終了した場合、上書きのために確保したCHUNK\_0031のストリームは上書きされなかったので開放される。この場合、特殊なタイトルの処理が行われる。すなわち、TITLE\_003の先頭から上書き記録を開始し、その途中で記録が終了した場合には、そこでタイトルが分割される。つまり、図35に示すように、上書き記録開始位置から終了位置までが新たなTITLE\_003とされ、それ以降の（元々のTITLE\_003の残り部分）はTITLE\_004とされる。

【0104】次に、タイトル再生の動作について説明する。いま、図26に示すようなファイルを有する光ディスク1を光ディスク装置に挿入し、タイトル再生するものとする。まず、光ディスク1が挿入されると、CPU21は情報ファイルを光ディスク1から読み込んで、RAM24に記憶させる。この動作は上述した、基本的な情報

ファイルの読み込み動作を繰り返すことで行われる。

【0105】CPU 21は、まず、VOLUME.TOCとALBUM.STRを読み出す。次にCPU 21は、ディレクトリ"TITLE"以下に、".VDR"の拡張子を持つファイルがいくつあるかを調べる。この拡張子を持つファイルは、タイトルの情報を持つファイルであり、そのファイルの数はつまりタイトルの数となる。図26の例ではタイトル数は3となる。次にCPU 21は3つのタイトル情報ファイルを読み込み、RAM 24に記憶させる。

【0106】CPU 21は、OSD制御回路9を制御して、光ディスク1上に記録されているタイトルの情報を示す文字情報を発生させ、合成回路8によりビデオ信号と合成させ、出力端子P1からディスプレイに出力させ、表示させる。いまの場合、タイトルが3つあること、そして3つのタイトルそれぞれの長さや属性(名前、記録された日時など)が表示される。

【0107】ここで、ユーザが、例えばTITLE\_002の再生を指定したとする。TITLE\_002の情報ファイルには(図16のtitle\_info()中のcgit\_file\_idには)、CHUNKGROUP\_001を指定するファイルIDが記録されており、CPU 21はこれを記憶するとともに、CHUNKGROUP\_001をRAM 24に格納させる。

【0108】次に、CPU 21は、TITLE\_002の開始時刻と終了時刻(図16のtitle\_info()中のtitle\_start\_chunk\_group\_time\_stampとtitle\_end\_chunk\_group\_time\_stamp)が、どのCHUNKに対応するかを調べる。これは、CHUNKGROUPの情報の中から、それぞれのCHUNKが登録されている情報(図23のchunk\_arrangement\_info()中のpresentation\_start\_cg\_time\_countとpresentation\_end\_cg\_time\_count)を比較することで行なわれる。いまの場合、図27に示すように、TITLE\_002の開始時刻は、CHUNK\_0001の途中に入っていることがわかる。つまり、TITLE\_002を先頭から再生するには、ストリームファイル"CHUNK\_0001.MPEG2"の途中から再生を開始すれば良いということがわかる。

【0109】次に、CPU 21は、TITLE\_002の先頭がストリーム中のどこにあたるかを調べる。すなわち、TITLE\_002の開始時刻が、ストリーム中のオフセット時刻(タイムスタンプ)としていくつにあたるのかが計算され、次にCHUNKファイル中の特徴点情報を使用して、開始時刻直前にあたる再生開始点が特定される。これにより、再生開始点のファイル先頭からのオフセット距離が確定できたことになる。

【0110】次に、CPU 21は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、"CHUNK\_0001.MPEG2"が記録されている光ディスク1上の物理アドレスと、その長さを確定する。更に、このアドレスに、先程求めた再生開始点のオフセットアドレスが加えられて、TITLE\_002の再生開始点のアドレスが最終的に確定される。

【0111】続いて、CPU 21は、この"CHUNK\_0001.MPEG2"のアドレス情報に基づき、光ヘッド2を読み出し位置に移動させる。そしてCPU 21は、光ヘッド2、RFおよび復調/変調回路3、並びにECC回路4を読み出しモードに設定するとともに、スイッチ5を読み出しチャンネル用バッファ6側に切り替え、光ヘッド2の位置を微調整した後、光ヘッド2による読み出しを開始させる。これにより"CHUNK\_0001.MPEG2"の内容が読み出しチャンネル用バッファ6に蓄積される。

【0112】読み出しチャンネル用バッファ6に蓄積されたデータは、デコーダ7に出力され、デコード処理が施されて、ビデオ信号とオーディオ信号が出力される。光ディスク1から読みだされ、デコードされ、表示されたデータ量が、"CHUNK\_0001.MPEG2"の大きさと等しくなった時点で、CPU 21は、TITLE\_003の再生に移行する。このTITLE\_003の再生動作は、TITLE\_002の再生動作と同様の動作である。

【0113】登録されているタイトルの再生が終了したとき、あるいは読み出し動作の停止が指示されたとき、読み出し、デコード処理が停止される。

【0114】なお、光ディスク装置に、光ディスク1として、新しいディスクが挿入された場合、あるいは、異なるフォーマットのディスクが挿入された場合、CPU 21は、ディスクが挿入されたとき、VOLUME.TOCとALBUM.STRを読み出そうとするが、これらのディスクには、このようなファイルが存在しないことになる。このような場合、即ち、VOLUME.TOCとALBUM.STRを読み出すことができない場合、CPU 21はメッセージを出力し、ユーザに指示を求める。ユーザは、CPU 21に指示し、光ディスク1をイジェクトさせるか(例えば、異なるフォーマットのディスクである場合)、初期化させるか(例えば、同一フォーマットの新しいディスクである場合)、または何らかの方法によりデータを復旧させる(例えば、同一フォーマットのディスクであるが、データが破壊されている場合)。

【0115】次に、titleについて、さらに説明する。図15に示すように、TITLE\_###.VDRは、titleの情報を格納するためのfileである。1つのtitleに関する情報は、1つのtitle\_info()に記録される。TITLE\_###.VDR内に存在するtitle\_info()の数は1個である。従って、volume内にはtitleの数だけTITLE\_###.VDRが存在する。

【0116】title numberは、図16のtitle\_info()の中で定義せず、file名またはfile idで決定される。従って、TITLE\_###.VDRのうちの正の整数###がtitle numberを表す。titleは構造というよりも、chunkgroupにつけられた、開始点を表すタイトルインデックスから、次のタイトルの先頭を表すタイトルインデックスまでの範囲、またはchunkgroupの終了点までの範囲の部分である。

【0117】図15のTITLE\_###.VDRのfile\_type\_id

は、図36に示すように、title\_info()が記録されたfileであることを示すidであり、長さ16の文字列で表される。text\_block()は、さまざまなtextを格納するための領域であり、そのtext\_block()で使用が許されているtext itemだけが記述される。

【0118】title\_info()は、図16に示すように、chunkgroup上における、titleの開始点と終了点、その他のtitleに関する属性が書かれる領域である。また、title\_info()は、タイトル番号順に再生したとき、タイトル間でシームレス再生が保証できるか否かを示すflagを持つことができる。このflagにより、光ディスク装置でタイトル間をシームレス再生できるか否かが事前に把握でき、また併合時に配置を変える必要があるかどうか分かる。

【0119】タイトル内およびチャンクグループ内ではシームレス再生が保証されているが、タイトルの境界はfileの境界でも有り得るため、タイトル間では、シームレス再生は保証されないことがある。ただし、光ディスク装置の機能として、再配置等を行うことにより、一般的にシームレス再生が行われるような状態にすることは可能である。

【0120】図16のtitle\_info()中のtitle\_info\_lengthは、title\_info()の長さをbyte単位で表したものである。flags\_for\_titleには、対応するtitleの書き込み属性(変更許可)、再生回数制限、ratingのlevel等が記録される。cgit\_file\_idには、対応するtitleのbaseであるchunkgroupのinformation file(CHUNKGROUP\_###.CGIT)のfile\_idが記録される。

【0121】title\_start\_chunk\_group\_time\_stampには、chunkgroupで定義されたlocalな時間軸上における、そのtitleの再生開始点の時刻が記録される。そのtitleのtitleindexが指しているピクチャの表示時刻がこの値となる。title\_end\_chunk\_group\_stampには、chunkgroupで定義されたlocalな時間軸上における、そのtitleの再生終了点の時刻が記録される。この値は、chunkgroupの再生終了点か、または時間軸上において直後に位置するtitleの開始点を表すtitleindexが示す値と同一となる。

【0122】title\_playback\_time()には、そのtitleの再生時間(タイムコード値、またはframe若しくはfield枚数)が記録される。number\_of\_marksには、そのtitle内に設定されているすべてのmarkの総数(titleindexを除く)が記録される。mark\_typeには、図37に示すように、title内の任意の位置につけられるmarkの種類が記録される。markは、title内のrandom access pointとしても利用される。mark\_chunk\_group\_time\_stampには、chunkgroupの時間軸上において、そのmarkが設定されている個所のtime stampが、値の小さなものから順に記録される。titleの開始点および終了点と同じtime stampを有するindexは存在してもよい。stuffing\_bytesに

は、stuffingするbytesが記録され、その長さは $8 \times n$  bit( $n \geq 0$ )となる。

【0123】次に、図21乃至図24に示したchunkgroupとchunkについてさらに説明する。CHUNKGROUP\_###.CGITは、titleの時間軸の定義、およびchunkの構成、titleに含まれる不連続点の処理を記述したファイルである。

【0124】titleは各種のbitstreamで構成されており、videoが無いstreamである場合や、DV(デジタルビデオ)のbitstreamである場合もある。DVフォーマットでは、フレーム単位で時間軸が規定されており、MPEG2 videoのSTC(System Time Clock)を基準にしていると、フォーマットが異なるので、このDVのbitstreamを管理することができない。

【0125】そこで、title内でlocalな時間軸を設定するものとする。この時間軸はtitleを構成するstreamに依存しない。titleの境界はchunkの境界とは無関係に設定される。そのため、localな時間軸は、chunk(bitstreamに1対1で対応される)毎や、title毎に設定するよりも、複数の(任意の数の)titleが含まれるchunkの集合体に対して設定するのが適切である。このchunkの集合体がchunkgroupである。

【0126】chunkgroupでは、単一の時間軸を定義し、その上にchunkを貼り付けていくことでchunkの表示時刻を定めている。つまり、chunkgroupには、bitstream fileの内容(byte列)を時間軸上に展開した状態でchunkが並ぶ。1つのbitstream fileに含まれるすべてのchunkを時間軸上に並べたものをpathと呼ぶ。chunkgroupには、複数のpathを並べることができる。pathのうち、chunkgroupの再生開始時刻と終了時刻を規定するpathはmain pathと称され、その他のpathは、sub pathと称される。sub pathは、主に、後から追加記録されたaudioのchunk等を表す。

【0127】chunkの接続点はtitleの境界とは必ずしも一致しないので、titleの属性ではない。しかし、chunk間の関係を各chunkの属性に含めると、階層的に矛盾が発生する。このような不連続点情報は、chunkとtitleの中間に位置するものであり、chunkgroupの階層に置くのが適切であると考えられる。

【0128】以上をまとめると、chunkgroupが持つ情報は、chunkの時間軸上への配置の仕方、chunkの再生順序、chunkの終わりと次に再生するchunkの始まりとの接続点で発生する不連続点などである。

【0129】なお、CHUNKGROUPは、同時に再生するSTREAMを指定することも可能である。例えば、図38に示す例では、TITLE\_001がCHUNKGROUP\_001を指定しており、CHUNKGROUP\_001はCHUNK\_0001とCHUNK\_0002を指定している。CHUNK\_0001で指定されるSTREAM\_0001と、CHUNK\_0002で指定するSTREAM\_0002は、それぞれ少なくともその一部が時間軸上で同時に重複して再生されるものとされて

いる。

【0130】図39は、このようなCHUNKGROUPが発生する場合の例を示している。この例においては、BITSTREAM Aを管理するCHUNK AとBITSTREAM Bを管理するCHUNK Bが合成され、CHUNKGROUPは、main pathとしてCHUNK Aを、また、sub pathとしてCHUNK Bを、それぞれ管理している。BITSTREAM AとBITSTREAM Bは、少なくともその一部が、それぞれ同時刻に重複して再生されるものである。

【0131】図21のCHUNKGROUP\_###.CGITのfile\_type\_idは、図40に示すように、そのファイルがCHUNKGROUP\_###.CGITであることを表す識別子であり、ISO 646に従った16文字の文字列で表される。chunkgroup\_time\_base\_flagsには、chunkgroupの基準カウンタに関するflagが記録される。chunkgroup\_time\_base\_offsetには、chunkgroup内の基準時間軸の開始時刻が記録される。この値は、90kHzのクロックをカウントアップするカウンタにセットされる値であり、64bitで表される。text\_block()は、さまざまなtextを格納するための領域であり、そのtext\_block()で使用が許されているtext itemだけが記述される。

【0132】図22に示すように、chunk\_connection\_info()は、特異な点の情報(videoの切り替え点、videoとaudioの同期など)を記録しておくためのファイルであり、chunk間の接続状況を規定している。編集によりできたchunkとchunkのつなぎ目のような特異点では、GOPの途中でchunkを乗りかえる必要がある。このような編集点付近の情報がここに記述される。chunkは2つ以上のchunkgroupに属することはない。

【0133】chunk\_connection\_info\_lengthには、chunk\_connection\_info()の長さをbyte単位で表したものが記録される。number\_of\_chunksには、そのchunkgroupで使われるchunkの総数が記録される。chunk\_sync\_play\_flagは、図41に示すように、同時刻に2つ以上のchunkを再生する必要があるかどうかを表すflagであり、その値の0は、1つのchunkの再生を意味し、その値の1は、複数のchunkの同時再生を意味する。

【0134】図23のchunk\_arrangement\_info()において、chunk\_arrangement\_info\_lengthには、各chunkについての情報の長さをbyte単位で表したものの(chunk\_arrangement\_info\_lengthの先頭byteからtransition\_info()の最終byteまでを含めた長さ)が記録される。chunk\_info\_file\_idには、対象となるchunk\_info\_fileのfile\_idが記録される。

【0135】chunk\_switch\_stream\_idには、2つのchunkを接続したとき連続再生するstreamのstream\_idが記録される。このidとしては、例えば、MPEG2のパケットヘッダに記録されているビデオあるいはオーディオを識別するidが用いられる。presentation\_start\_cg\_time\_countには、該当chunkの表示開始時刻を、chunkgroup内の

時刻で表したtime count値が記録される。chunkの表示開始時刻は、chunkgroup内で定義されるglobalなtime stampで表現される。該当chunkは、chunkgroupにおいて、この時刻から表示が開始される。presentation\_end\_cg\_time\_countには、該当chunkの表示終了時刻を、chunkgroup内の時刻で表したtime count値が記録される。chunkの表示終了時刻は、chunkgroup内で定義されるglobalなtime stampで表現される。

【0136】図42に示すように、original\_time\_count\_typeには、stream内部で使われているtime countの種類が記録される。例えば、MPEG2 videoのstreamであれば、original\_time\_count\_typeは'0000'とされる。number\_of\_start\_original\_time\_count\_extensionには、複数のtime countが必要なとき、新たに必要な開始時刻を表すtime countの数が記録される。number\_of\_end\_original\_time\_count\_extensionには、複数のtime countが必要なとき、新たに必要な終了時刻を表すtime countの数が記録される。presentation\_start\_original\_time\_countには、presentation\_start\_cg\_time\_countに対応する、stream内部における時刻あるいはカウンタ値が記録される。presentation\_end\_original\_time\_countには、presentation\_end\_cg\_time\_countに対応する、stream内部における時刻あるいはカウンタ値が記録される。

【0137】tc\_ext\_attributesには、time\_count\_extensionに対する属性が記録される。このtime\_count\_extensionには、例えば、どのstreamに適用するのか等の情報を入れることができる。start\_original\_time\_count\_extensionには、chunkの切り替えに必要な開始時刻あるいは開始カウンタ値が記録される。これはオプションであり、複数の時刻やカウンタ値を記録する必要があるときに使用される。end\_original\_time\_count\_extensionには、chunkの切り替えに必要な終了時刻あるいは終了カウンタ値が記録される。これもオプションであり、複数の時刻やカウンタ値を記録する必要があるときに使用される。transition\_info()には、chunkの切り替えて特殊効果をかけるときに必要な情報が記録される。例えば、chunkの指定、切り替え時刻、特殊効果の種類等がここに記述される。

【0138】図24に示すように、CHUNK\_%%%.ABSTは、sub\_file番号%%%のchunkを構成するbitstreamから抽出した特徴点を記録したfileである。このfileには、GOP、Audio frame等のbitstreamを構成する単位毎に、その開始byte位置、長さ、属性等が記述される。GOP情報、Audio frame情報はchunk(sub-file)毎に、各々1つのCHUNK\_%%%.ABSTとしてまとめられる。

【0139】図43に示すように、CHUNK\_%%%.ABSTのfile\_type\_idには、stream\_info()が記録されているfileであることを示す識別子が、ISO 646に従った16文字の文字列で記録される。

【0140】図44に示すように、info\_typeには、図24において次に続くstream\_infoのtypeが記録される。ここでstreamの種類が特定される。number\_of\_programsには、MPEG2のTS(Transport Stream)に含まれるprogramの数が記録される。この数を知るには、PSI(Program Specific Information)を読み取る必要がある。TS以外のときには、この値は1になる。number\_of\_streamsには、そのprogramで使われるストリームの数が記録される。この値は、TSの場合には、異なるPID(packet identification)の数と等しくなる。TS以外のMPEGstreamの場合、ここには、stream idが異なるstreamの数が記録される。

【0141】stream\_identifierには、stream idあるいは、stream idを拡張したものが記録される。TSの場合には、stream idとして、PIDが利用される。

【0142】図45に示すように、slot\_unit\_typeには、streamをある一定間隔毎に区切ったときの、その区切り方が記録される。各frame、field等、区切りの指標が時間の場合には、time stamp valueが用いられる。slot\_time\_lengthには、1 slotに対応する時間が記録される。この値は、90kHzのクロックをカウントするカウンタを用いたtime stampの値で表される。number\_of\_slotsには、CHUNK\_%%.ABSTに書かれているslot\_info()の数が記録される。number\_of\_I\_pictures\_in\_a\_slotには、slotに含まれるI-pictureの数が記録される。この値は、1以上の整数で15以下の値となる。ただし、GOPheaderを先頭とするslotの直前に位置するslotに含まれるI-pictureの数は、この値よりも小さくてもよい。GOPheaderの直後でないI-pictureのpicture headerを先頭とするslotを設定するとき、この値が活用される。

【0143】次に、図17と図18に示したprogramについてさらに説明する。PROGRAM\_\$\$\$PGIには、program()がただ1つ存在する。volume内にはprogramの数だけPROGRAM\_\$\$\$PGIが存在する。program番号は、program()の中で定義せず、file名またはfile idで規定される。

【0144】図46に示すように、図17のPROGRAM\_\$\$\$PGIのfile\_type\_idには、program()が記録されたfileであることを示すidが、長さ16の文字列で記録される。text\_block()には、さまざまなtextを格納するための領域が形成されている。ここには、そのtext\_block()で利用が許されているtext itemだけが記述される。

【0145】図18のprogram()のflags\_for\_programには、programに関する各種フラグが記録される。例えば、このprogramの書き込み属性(変更許可)、再生回数制限、ratingのlevel等が記録される。

【0146】図47に示すように、program\_statusには、programの属性が記録される。このfieldの設定はoptionであるが、設定しないときは"none"にしなければならない。

【0147】program\_playback\_time()には、そのprogramの再生時間が記録される。

number\_of\_play\_sequencesには、そのprogramで使用されているplay\_sequenceの数が記録される。ただし、このformatの例では、値は1に固定されている。すなわち、このformatの例では、1 program=1ch(チャンネル)再生とされているので、2ch同時再生を実現するには、2programの同時再生指定を可能にすればよい。1program=1ch再生の制限がなければ、1programで、2ch同時再生も可能である。multichannel I/Oを使って2つのplay sequenceを同時再生するとき、どのplay sequenceをどのoutput channelに割り当てるかは、光ディスク装置が決める。

【0148】number\_of\_play\_listsには、このplay sequenceで使用されているplay\_listの数が記録される。この例では値は1とされる。play\_list\_start\_time\_stamp\_offsetには、play sequenceの開始時刻から開始するtimerでカウントした、play sequence内での時刻が記録される。この値が、play listの開始時刻になる。programでは、play sequence内にplay listは1つしか存在してはならない。時刻の単位系は90kHzである(1/90000秒が時刻の最小単位である)。stuffing\_bytesには、stuffingのbytesが記録される。その長さは、 $8 \times n$  bit( $n \geq 0$ )とされる。

【0149】次に、タイトルを分割したり、移動したりするタイトル編集処理について説明する。タイトル分割は、現存するタイトルをユーザの指定した位置で分割し、新たなタイトルとするものである。また、タイトル移動とは、タイトルの順番を入れ替えることを意味する。上述したフォーマット構造のタイトル情報ファイルは、これらの分割や移動を簡単に実行することができるような構造とされている。すなわち、このフォーマットにおいては、ビットストリームファイルをまとめた構造体(チャンクグループ)と、ユーザがタイトルとして認識する構造体(タイトル)が分割されており、これにより、タイトルを分割したり、移動したりしても、チャンクグループ以下の情報を変更する必要がなくなるようになされている。

【0150】次に、図48のフローチャートを参照して、タイトル分割処理について説明する。最初にステップS1において、ユーザは、分割点を指示する。例えば、図27に示すTITLE\_002の所定の位置を分割点として指定したものとする。このとき、ステップS2において、CPU21は、図49に示すように、分割対象タイトル(いまの場合、TITLE\_002)の開始点を開始点とし、分割点を終了点とする第1のタイトルの情報ファイル(TITLE\_002.VDR)を作成する。

【0151】次に、ステップS3において、CPU21は、分割対象タイトル(分割前のTITLE\_002)の分割点を開始点とし、終了点を終了点とする第2のタイトルの情報ファイル(TITLE\_003.VDR)を作成する。

【0152】次に、ステップS4において、CPU21

は、分割対象タイトルより後のタイトルの名称を変更する。すなわち、図49に示すように、図27において、TITLE\_003とされていたものが、TITLE\_004とされる。

【0153】このように、分割処理を行っても、CHUNKGROUP、CHUNK、STREAMについては、一切変更が加えられない。

【0154】図50は、タイトルの入れ替え処理を表している。この例では、図49に示すTITLE\_002と、TITLE\_003を入れ替えるものとする。

【0155】この場合、ステップS11において、CPU21は、元の番号002のtitle (TITLE\_002) の名称を番号Xのtitle (TITLE\_X) に変更する。次に、ステップS12において、CPU21は、元の番号003のtitle (TITLE\_003) の名称を番号002のtitle (TITLE\_002) に変更する。さらに、ステップS13において、CPU21は、番号Xのtitle (TITLE\_X) を番号003のtitle (TITLE\_003) に変更する。

【0156】このように、番号002のtitle (TITLE\_002) を番号Xのtitle (TITLE\_X) に一旦変更した後、さらに番号003のtitle (TITLE\_003) に変更するのは、番号002のtitle (TITLE\_002) を番号003のtitle (TITLE\_003) に直接変更すると、その時点において、同一の番号003のtitle (TITLE\_003) が2個生成された状態になるので、これを避けるためである。

【0157】このようにして、図51に示すように、番号002のtitle (TITLE\_002) と番号003のtitle (TITLE\_003) が入れ替えられたことになる。

【0158】図52は、タイトル削除の処理を説明するフローチャートである。ステップS21において、ユーザが削除するtitleを指定すると、ステップS22において、CPU21は、指定されたtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMが、他のtitleでも対応されているか否かを判定する。他のtitleによって対応されていない場合には、ステップS23に進み、CPU21は、削除を指定されたtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMを削除する。

【0159】ステップS22において、削除を指定されたtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMが、他のtitleによっても対応されていると判定された場合には、これを削除すると、他のtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMが存在しなくなってしまうことになる。そこで、この場合には、ステップS23の処理はスキップされる。

【0160】ステップS23の処理の後、または、ステップS22において削除を指定されたtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMが、他のtitleでも対応されていると判定された場合、ステップS24に進み、CPU21は、削除されずに残った情報ファイルを処理する。すなわち、titleが削除されたので、それより後の番号のtitleを1つずつ繰り上げる処理を実行す

る。ただし、ステップS22において削除を指定されたtitleが対応するCHUNKGROUP、CHUNK、およびSTREAMが、他のtitleでも対応されていると判定された場合には、実質的には削除が行われないので、OS制御回路9から実質的には削除が行われていない旨のメッセージが出力され、表示される。

【0161】このようにして、図53に示すように、TITLE\_002を削除すると、それまでのTITLE\_003がTITLE\_002とされ、それまでのTITLE\_004がTITLE\_003とされる。同様に、それまでのCHUNKGROUP\_001は、CHUNKGROUP\_001とCHUNKGROUP\_002に分割され、それまでのCHUNKGROUP\_002は、CHUNKGROUP\_003とされる。さらにそれまでのSTREAM\_001が、STREAM\_001とSTREAM\_002に分割される。

【0162】以上はタイトル削除に際し、STREAMを削除して空き領域を増やす例を示した。一方、管理情報であるtitle情報だけを削除し、実体であるSTREAMの削除は行わない方法もある。この場合には、タイトル削除を行っても、CHUNKGROUP、CHUNK、STREAMについては一切変更が加えられない。

【0163】次に、図54のフローチャートを参照して、タイトルを併合する処理について説明する。この場合、ステップS31において、ユーザは、併合するtitleを指定する。例えば、図49に示すTITLE\_002とTITLE\_003を併合するものとして指定すると、ステップS32において、CPU21は、前側のtitle (TITLE\_002) の開始点を開始点とし、後ろ側のtitle (TITLE\_003) の終了点を終了点とするtitleを作成する。いまの場合、このtitleは、TITLE\_002とされる。

【0164】次に、ステップS33に進み、古いtitleを削除する処理が実行される。すなわち、この場合、元のTITLE\_002とTITLE\_003が削除される。さらにステップS34において、CPU21は、併合されたtitleより後のtitleの情報ファイルを処理する。すなわち、この例の場合、図49のTITLE\_004がTITLE\_003に変更される。このようにして、図27に示すようなtitleが得られることになる。

【0165】なお、以上の例では、2つのtitleが同一のCHUNKGROUPに属しており、相互に隣接している場合を例としたが、2つのtitleの専有するCHUNKGROUP、CHUNK、およびSTREAMを併合する順序に並べて、1つのCHUNKGROUPを構成するように併合することも可能である。

【0166】次に、programについて説明する。program再生は、素材（ストリーム）に対して、不可逆な編集を施すことなく、titleの必要な部分を集めて再生することを意味する。これを実現するための構造が上述したprogramである。このprogram再生は、ポイント再生とも称される。

【0167】programは、複数のplay sequenceから構成され、各play sequenceは、複数のplay listから構成される。1つのplay sequenceは、1つのoutput channel

を制御する。光ディスク装置が有しているoutput channelのどれにplay sequenceを割り当てるかは、光ディスク装置の仕様により決定される。

【0168】play sequenceを構成するplay listは、play item間でデコードまたは表示の時間的重なりを表現するためにある。play list内では、play itemが、その処理が時間的に重なることがないように配列されている。従って、play listの内部では、シーケンシャルにplay itemが処理されることになる。

【0169】但し、このフォーマット例では、programは、1つのplay sequenceから構成され、play sequenceは、1つのplay listから構成されているものとする。

【0170】play listは、表示時刻に重なりのないplay itemの並びで構成される。programでは、play itemとしてtitle中の領域を差し示すイン点とアウト点の組を有する。play itemのつなぎ目では、シームレス再生になることもあるが、ならないこともある。すなわち、シームレス再生は保証されていない。

【0171】次に、図55のフローチャートを参照して、プログラム再生の設定処理について説明する。いま、例えば図27に示すように、PROGRAM\_001が存在する状態において、さらに別のPROGRAM\_002を生成するものとする。

【0172】最初に、ステップS41において、ユーザは、プログラム再生するtitleと、そのイン点およびアウト点を指定する。例えば、図56に示すように、TITLE\_003の所定の第1の位置をイン点と指定し、所定の第2の点をアウト点として指定する。

【0173】次に、ステップS42に進み、CPU21は、program() (図18) のplay\_list() (図19) のplay\_item() (図20) のtitle\_numberに、ステップS41で指定されたtitle (いまの場合、TITLE\_003) を設定し、item\_start\_time\_stampに、ステップS41で指定されたイン点を設定し、item\_end\_time\_stampに、ステップS41で指定されたアウト点を設定する。

【0174】次に、ステップS43において、CPU21は、program()を含むファイルをPROGRAM\_\$\$\$ .PGIとして、光ディスク1に記録する。なお、ここで、\$\$\$は、program番号とされる。すなわち、いまの場合、\$\$\$は、002とされる。

【0175】以上のようにして、図56に示すように、PROGRAM\_002が作成される。

【0176】次に、図57のフローチャートを参照して、プログラム再生処理について説明する。ユーザが、図27または図56に示すような情報が記録されている光ディスク1を光ディスク装置に装着したとき、ステップS51において、CPU21は、光ディスク1に対して、上述した「基本的な情報ファイル読み込み動作」を繰り返し、情報ファイルを読み出し、RAM24に記憶させる。

【0177】すなわち、最初に、CPU21は、VOLUME.TOCとALBUM.STRを読み出し、次に、ファイルシステムに対し、ディレクトリ“program”以下に、“ .PGI” の拡張子を有するファイルがいくつあるかを問い合わせる。この拡張子を持つファイルは、プログラム再生ファイルの情報を持つファイルであり、そのファイルの数がprogram再生の数となる。図27の例の場合、この数は1となり、図56の例の場合、この数は2となる。

【0178】次に、ステップS52に進み、CPU21は、OSD制御回路9を制御し、プログラム再生の数、長さ、属性などをRAM24から読み出し、ディスプレイに表示させる。ここで、属性とは、例えば、名前、記録された日時などである。

【0179】次に、ステップS53において、ユーザは、再生するプログラムを指定する。例えば、図56の例では、programが2個存在するので、そのうちのいずれを再生するのかをユーザは入力部14を操作して指定する。ユーザが再生するプログラムを指定すると、ステップS54において、CPU21は、指定されたプログラムを再生する処理を実行する。

【0180】ステップS54において、プログラム再生が実行されると、イン点で指定された位置から、アウト点で指定された位置までが再生される。例えば、図56に示す例では、PROGRAM\_001を再生するとき、TITLE\_001のイン点からアウト点までの範囲と、TITLE\_003のイン点からアウト点までの範囲が再生される。PROGRAM\_002が再生される場合には、TITLE\_003のイン点からアウト点までの範囲が再生される。

【0181】ステップS55においては、CPU21は、プログラムの再生が終了したか否かを判定し、終了していない場合には、ステップS54に戻り、それ以降の処理を繰り返し実行する。ステップS55において、プログラムの再生が終了したと判定された場合、ステップS56に進み、CPU21は、他にも再生すべきプログラムが存在するか否かを判定し、存在する場合には、ステップS53に戻り、それ以降の処理を繰り返し実行する。ステップS56において、全てのプログラムを再生したと判定された場合、処理は終了される。

【0182】プログラム (program)、プレイシーケンス (play sequence)、およびプレイアイテム (play item) の関係を示すと、図58乃至図60に示すようになる。

【0183】すなわち、プログラムは、図58に示すように、複数のプレイシーケンスを所定の順序で組み合わせることで構成される。そして、各プレイシーケンスは、図59に示すように、任意の数のプレイリストにより構成される。このプレイリストには、任意の数のプレイアイテムが、所定の時刻 (タイムスタンプ) に対応して記述されている。

【0184】図59の例においては、プレイリスト1

に、プレイアイテム1、プレイアイテム2、プレイアイテム3、プレイアイテム6、およびプレイアイテム7が、順次再生されるように記述されている。プレイリスト2には、プレイアイテム4が、プレイアイテム2の途中から再生が開始され、プレイアイテム3の途中で再生が終了するタイミングで記述されている。さらにプレイリスト3には、プレイアイテム5が、プレイアイテム3とプレイアイテム4の途中で再生が開始され、プレイアイテム6の途中で再生が終了されるように記述されている。従って、このプレイシーケンスが再生されると、プレイアイテム1、プレイアイテム2、プレイアイテム3、プレイアイテム6、およびプレイアイテム7が、順次連続して再生されるとともに、プレイアイテム4が、プレイアイテム2の途中からプレイアイテム3の途中まで同時に再生され、また、プレイアイテム3とプレイアイテム4が再生されている途中で、プレイアイテム5も同時に再生が開始され、プレイアイテム6の途中まで再生が継続される。

【0185】プログラム、プレイシーケンス、およびプレイアイテムの関係をまとめて図示すると、図60に示すようになる。すなわち、プログラムは、任意の数のプレイシーケンスにより構成され、プレイシーケンスは、任意の数のプレイリストにより構成される。そして、各プレイリストは、任意の数のプレイアイテムにより構成される。

【0186】以上においては、本発明を光ディスク装置に応用した場合を例として説明したが、本発明は、その他の記録媒体に情報を記録または再生する場合にも適用することが可能である。

【0187】なお、上記したような処理を行うコンピュータプログラムをユーザに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することができる。

【0188】

【発明の効果】以上の如く、請求項1に記載の情報処理装置によれば、任意の数の第1の管理手段を管理する第2の管理手段の任意の範囲を第3の管理手段で管理するようにした。また、請求項5に記載の情報処理方法、および請求項6に記載の提供媒体によれば、第1の管理ステップでの任意の数の管理状態を管理する第2の管理ステップでの任意の範囲の管理状態を第3の管理ステップで管理するようにした。さらに、請求項7に記載の記録媒体によれば、情報を管理する管理情報として、第1の管理手段乃至第3の管理手段を記録媒体に記録するようにした。従って、いずれの場合においても、簡単に、短時間で、編集処理を実行することが可能となる。

【図面の簡単な説明】

【図1】ディレクトリの構造を説明する図である。

【図2】VOLUME.TOCを説明する図である。

【図3】volume\_information()を説明する図である。

【図4】volume\_attribute()を説明する図である。

【図5】resume()を説明する図である。

【図6】volume\_rating()を説明する図である。

【図7】write\_protect()を説明する図である。

【図8】play\_protect()を説明する図である。

【図9】recording\_timer()を説明する図である。

【図10】text\_block()を説明する図である。

【図11】language\_set()を説明する図である。

【図12】text\_item()を説明する図である。

【図13】ALBUM.STRを説明する図である。

【図14】album()を説明する図である。

【図15】TITLE\_###.VDRを説明する図である。

【図16】title\_info()を説明する図である。

【図17】PROGRAM\_\$\$\$ .PGIを説明する図である。

【図18】program()を説明する図である。

【図19】play\_list()を説明する図である。

【図20】play\_item()を説明する図である。

【図21】CHUNKGROUP\_###.CGITを説明する図である。

【図22】chunk\_connection\_info()を説明する図である。

【図23】chunk\_arrangement\_info()を説明する図である。

【図24】CHUNK\_%%%.ABSTを説明する図である。

【図25】本発明を適用した光ディスク装置の構成例を示すブロック図である。

【図26】ディレクトリの構造を説明する図である。

【図27】ディレクトリの論理構造を説明する図である。

【図28】offsetを説明する図である。

【図29】ディレクトリの構造を説明する図である。

【図30】ディレクトリの構造を説明する図である。

【図31】ディレクトリの論理構造を説明する図である。

【図32】ディレクトリの構造を説明する図である。

【図33】ディレクトリの構造を説明する図である。

【図34】ディレクトリの論理構造を説明する図である。

【図35】ディレクトリの論理構造を説明する図である。

【図36】file\_type\_idを説明する図である。

【図37】mark\_typeを説明する図である。

【図38】CHUNKGROUPを説明する図である。

【図39】CHUNKGROUPを説明する図である。

【図40】file\_type\_idを説明する図である。

【図41】chunk\_sync\_play\_flagを説明する図である。

【図42】original\_time\_count\_typeを説明する図である。

【図43】file\_type\_idを説明する図である。

【図44】info\_typeを説明する図である。

【図45】 slot\_unit\_typeを説明する図である。

【図46】 file\_type\_idを説明する図である。

【図47】 program\_statusを説明する図である。

【図48】 タイトル分割処理を説明するフローチャートである。

【図49】 PROGRAM, TITLE, CHUNKGROUP, CHUNK, STREAMの階層を説明する図である。

【図50】 タイトルの入れ替え処理を説明するフローチャートである。

【図51】 PROGRAM, TITLE, CHUNKGROUP, CHUNK, STREAMの階層を説明する図である。

【図52】 タイトル削除処理を説明するフローチャートである。

【図53】 タイトル削除処理を説明する図である。

【図54】 タイトルの併合処理を説明するフローチャートである。

【図55】 プログラム再生の設定処理を説明するフローチャートである。

【図56】 PROGRAM, TITLE, CHUNKGROUP, CHUNK, STREAMの階層を説明する図である。

【図57】 プログラム再生処理を説明するフローチャートである。

【図58】 プログラムの構成を説明する図である。

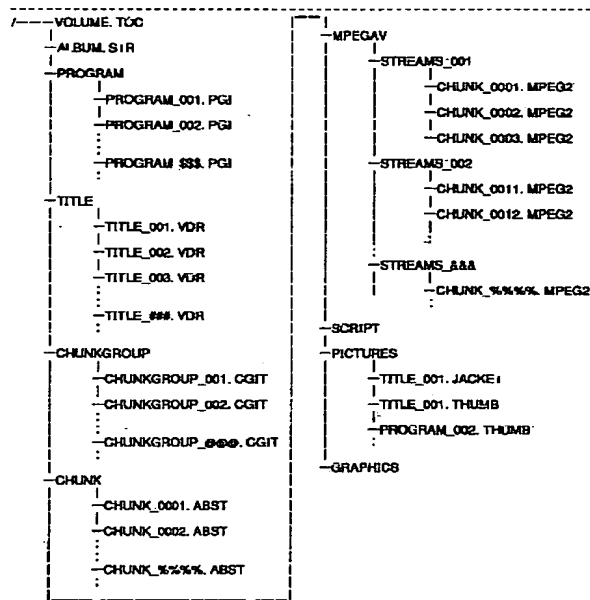
【図59】 プレイシーケンスの構成を説明する図である。

【図60】 プログラム、プレイシーケンス、およびプレイアイテムの関係を説明する図である。

【符号の説明】

1 光ディスク, 2 光ヘッド, 3 RFおよび復調/変調回路, 4 ECC回路, 6 読み出しチャンネル用バッファ, 7 デコーダ, 8 合成回路, 9 OSD制御回路, 10 エンコーダ, 11 書き込みチャンネル用バッファ, 12 アドレス検出回路, 13 システムコントローラ, 14入力部, 21 CPU, 22 ROM, 23, 24 RAM

【図1】



【図3】

Syntax	Number of Bits	Mnemonic
volume_information () {		
volume_attribute ()		
resume ()		
volume_rating ()		
write_protect ()		
play_protect ()		
recording_timer ()		
}		

【図2】

Syntax	Number of Bits	Mnemonic
VOLUME.TOC {		
file_type_id	8*16	char[16]
volume_information ()		
text_block ()		
}		

【図4】

Syntax	Number of Bits	Mnemonic
volume_attribute () {		
volume_attribute_length	32	ulmsbf
vdr_version	4*4	bod
reserved	6	bslbf
title_playback_mode_flag	1	bslbf
program_playback_mode_flag	1	bslbf
volume_play_time ()	4*8	bod
update_time_count ()	32	ulmsbf
maker_id	8*16	char[16]
model_code	8*16	char[16]
POSID	32	bslbf
}		

【図7】

Syntax	Number of Bits	Mnemonic
write_protect () {		
write_protect_length	32	ulmsbf
volume_write_protect_level	4	ulmsbf
password_enable_flag	1	bslbf
append_only_flag	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_write_protect	128	bslbf
reserved	8	bslbf
write_protect_set_time ()	56	bod
reserved	8	bslbf
write_protect_expiration_time ()	56	bod
number_of_times	18	ulmsbf
}		

【図5】

Syntax	Number of Bits	Mnemonic
resume () {		
resume_length	32	uimsbf
reserved // for byte alignment	3	bslbf
resume_switch	1	bit
reserved	4	bslbf
number_of_records	4	uimsbf
reserved // for byte alignment	7	bslbf
resume_auto_executes_time_flag	1	bit
resume_auto_executes_time ()	4*14	bcd
reserved	4	bslbf
resume_auto_executes_record_number	4	uimsbf
for (i=0; i<number_of_records; i++) {		
resume_mode_flag	4	bslbf
object_type	4	bslbf
linked_record_number	4	uimsbf
number_of_times	18	uimsbf
resume_updated_time ()	4*14	bcd
switch (object_type) {		
case title :		
title_number	16	uimsbf
title_local_time_stamp	64	uimsbf
break ;		
case program :		
program_number	18	uimsbf
program_local_time_stamp	64	uimsbf
break ;		
case program_bind :		
program_bind_number	16	uimsbf
program_order	16	uimsbf
program_number	18	uimsbf
program_local_time_stamp	64	uimsbf
break ;		
case play_item :		
play_item_number	16	uimsbf
play_item_local_time_stamp	64	uimsbf
break ;		
}		
}		
}		

【図8】

Syntax	Number of Bits	Mnemonic
play_protect () {		
play_protect_length	32	uimsbf
volume_play_protect_flag	2	bslbf
reserved	2	bslbf
password_enable_flag	1	bslbf
reserved	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_play_protect	128	bslbf
reserved	8	bslbf
play_protect_set_time ()	56	bcd
reserved	8	bslbf
play_protect_expiration_time ()	56	bcd
number_of_times	16	uimsbf
}		

【図10】

Syntax	Number of Bits	Mnemonic
text_block () {		
text_block_length	32	uimsbf
number_of_language_sets	8	uimsbf
number_of_text_items	16	uimsbf
for (i=0; i<number_of_language_sets; i++) {		
language_set ()		
}		
for (i=0; i<number_of_text_items; i++) {		
text_item ()		
}		
}		

【図29】

```

-----MPEGAV
|
|-----STREAMS_003
|
|-----CHUNK_0031.MPEG2

```

【図6】

Syntax	Number of Bits	Mnemonic
volume_rating () {		
volume_rating_id	8*18	char[18]
volume_rating_length	32	uimsbf
reserved	6	bslbf
volume_rating_type	2	bslbf
volume_rating_password	128	bslbf
switch (volume_rating_type) {		
case age_limited :		
number_of_rating	8	uimsbf
for (i=0; i<number_of_rating; i++) {		
country_code_for_rating	24	bslbf
age_for_volume_rating	8	uimsbf
}		
break ;		
case CARA :		
reserved	4	bslbf
CARA_category	4	bslbf
reserved	24	bslbf
break ;		
case HSAC :		
reserved	4	bslbf
HSAC_category	4	bslbf
reserved	4	bslbf
RSAC_level	4	bslbf
reserved	16	bslbf
break ;		
}		
}		

【図9】

Syntax	Number of Bits	Mnemonic
recording_timer () {		
recording_timer_length		
recording_timer_flag		
number_of_entry		
for (i=0; i<number_of_entry; i++) {		
date_and_time		
channel		
program		
}		
}		

【図11】

Syntax	Number of Bits	Mnemonic
language_set () {		
reserved	8	bslbf
language_code	24	bslbf
character_set_type	8	bslbf
number_of_language_set_names	8	uimsbf
for (i=0; i<number_of_language_set_names; i++) {		
character_set_type_for_name	8	bslbf
language_set_name_length	8	uimsbf
language_set_name	8*language_set_name_length	bslbf
}		
}		

【図12】

Syntax	Number of Bits	Mnemonic
text_item () {		
text_item_length	18	uimsbf
text_item_id	18	uimsbf
text_item_sub_id	16	uimsbf
flags	8	bslbf
number_of_used_language_sets	8	uimsbf
// loop for each language set		
for (i=0; i<number_of_used_language_sets; i++) {		
language_set_id	8	uimsbf
reserved	4	bslbf
text_string_length	16	uimsbf
text_string	8*text_string_length	bslbf
bitmap ()		
}		
stuffing_bytes	8*n	bslbf
}		

【図13】

Syntax	Number of Bits	Mnemonic
ALBUM_### { file_type_id album_id text_block () }	8*16	char[16]

【図15】

Syntax	Number of Bits	Mnemonic
TITLE_### VDR { file_type_id title_info () text_block () }	8*16	char[16]

【図21】

Syntax	Number of Bits	Mnemonic
CHUNKGROUP_### CGIT { file_type_id  chunkgroup_time_base_flags chunkgroup_time_base_offset  chunk_connection_info () text_block () }	8*16  32 64	char[16]  bslbf ulmsbf

【図16】

Syntax	Number of Bits	Mnemonic
title_info () { title_info_length flags_for_title cgit_file_id title_start_chunk_group_time_stamp title_end_chunk_group_time_stamp  title_playback_time () reserved number_of_marks for (l=0; l<number_of_marks; l++) { reserved mark_type mark_chunk_group_time_stamp } stuffing_bytes }	32 32 16 64 64  32 32 16  4 4 64 8*n	ulmsbf bslbf ulmsbf ulmsbf ulmsbf  bcd bslbf ulmsbf  bslbf bslbf ulmsbf bslbf

【図18】

Syntax	Number of Bits	Mnemonic
program () { program_length flags_for_program program_status program_playback_time () reserved  number_of_play_sequences for (j=0; j<number_of_play_sequences; j++) { number_of_play_lists for (k=0; k<number_of_play_lists; k++) { play_list_start_time_stamp_offset play_list (k) } } stuffing_bytes }	32 32 4 32 32  16 16 64 8*n	ulmsbf bslbf bslbf bcd bslbf  ulmsbf ulmsbf ulmsbf bslbf

【図14】

Syntax	Number of Bits	Mnemonic
album () { album_length reserved volume_status if (volume_status == "1b") { chief_volume_flag } else { reserved }  if (volume_status == "1b") { if (chief_volume_flag == "1b") { reserved album_type album_id  number_of_discs_in_album number_of_volumes_in_album  for (l=0; l<number_of_volumes_in_album; l++) { disc_id_for_album_member volume_id_for_album_member title_offset_number } reserved_for_program_bind number_of_program_binds for (l=0; l<number_of_program_binds; l++) { number_of_program_in_this_program_bind for (l=0; l<number_of_program_in_this_program_bind; l++) { disc_id_for_program_bind_member volume_id_for_program_bind_member program_number } } } else { // chief_volume_flag == "0b" chief_disc_id chief_volume_id (album_id) } } }	32 6 1 1 1 1  6 2 128  16 16  128 128 16  8 8 16 128 128 16 128 128 128	ulmsbf bslbf bslbf bslbf bslbf "0"  bslbf bslbf bslbf  ulmsbf ulmsbf  bslbf bslbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf ulmsbf

【図17】

Syntax	Number of Bits	Mnemonic
PROGRAM_### PGI { file_type_id program () text_block () }	8*16	char[16]

【図19】

Syntax	Number of Bits	Mnemonic
play_list () { // playback sequence of play items in this play list number_of_play_items for (k=0; k<number_of_play_items; k++) { play_item_number reserved seamless_connection_flag }  // play_item_table for (PIN=1; PIN<=number_of_play_items_in_program; PIN++) { play_item () } }	16  16 31 1	ulmsbf  ulmsbf bslbf bslbf

【図22】

Syntax	Number of Bits	Mnemonic
chunk_connection_info () { chunk_connection_info_length reserved number_of_chunks chunk_sync_play_flag  // chunk info file list for (l=0; l<number_of_chunks; l++) { chunk_arrangement_info () } }	32 16 16 8	ulmsbf bslbf ulmsbf bslbf

【図20】

Syntax	Number of Bits	Mnemonic
play_item () {		
play_item_length	32	uint32
play_item_type	8	bsb1
play_mode	8	bsb1
total_playback_time ()	32	uint32
}		
menu_item_number	18	uint18
return_item_number	18	uint18
next_item_number	18	uint18
prev_item_number	18	uint18
if (play_item_type == "0000b") {		
// play item for one "cut"		
title_number	18	uint18
// IN point		
item_start_time_stamp	64	uint64
// OUT point		
item_end_time_stamp	64	uint64
}		

【図24】

Syntax	Number of Bits	Mnemonic
CHUNK %%% ABSI {		
file_type_id	8*18	char[18]
reserved	4	bsb1
chunk_file_id	18	uint18
info_type	4	bsb1
// stream_info ()		
if (info_type == "MPEG2_System_TS") {		
number_of_programs	8	uint8
} else {		
number_of_programs	8	"0000 0001"
} for (i=0; i<number_of_programs; i++) {		
number_of_streams	8	uint8
for (j=0; j<number_of_streams; j++) {		
stream_identifier	16	bsb1
// slot type information		
reserved	4	bsb1
slot_unit_type	4	bsb1
if (slot_unit_type == "time_stamp") {		
slot_time_length	32	uint32
} else {		
reserved	32	bsb1
} number_of_slots	32	uint32
reserved	4	bsb1
switch (info_type) {		
case MPEG1_System:		
case MPEG2_System_PS:		
case MPEG2_System_TS:		
case video_elementary_stream	4	uint4
number_of_I_pictures_in_a_slot		
break;		
default:	4	bsb1
reserved		
break;		
}		
// stream attribute		
IS_attribute ()		
}		
} // loop of slot info		
for (i=0; i<number_of_streams; i++) {		
for (j=0; j<number_of_slots; j++) {		
slot_info ()		
}		
}		

【図32】

```

----- MPEGAV
|         |
|         |-----STREAMS_002
|         |         |
|         |         |-----CHUNK_0031. MPEG2

```

【図23】

Syntax	Number of Bits	Mnemonic
chunk_arrangement_info () {		
chunk_arrangement_info_length	32	uint32
chunk_info_id	16	bsb1
reserved	5	bsb1
chunk_switch_stream_id	16	bsb1
presentation_start_cg_time_count	64	uint64
presentation_end_cg_time_count	64	uint64
reserved	4	bsb1
chunk_time_count_type	4	bsb1
number_of_start_original_time_count_extension	8	uint8
number_of_end_original_time_count_extension	8	uint8
// presentation start position and time		
presentation_start_original_time_count	64	uint64
presentation_end_original_time_count	64	uint64
for (j=0; j<number_of_start_original_time_count_extension; j++)		
to_ext_attributes	16	bsb1
start_original_time_count_extension	64	uint64
}		
// presentation end position and time		
for (k=0; k<number_of_end_original_time_count_extension; k++) {		
to_ext_attributes	16	bsb1
end_original_time_count_extension	64	uint64
}		
transition_info ()		

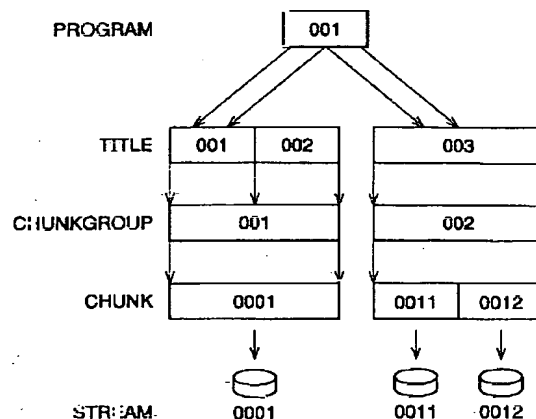
【図26】

```

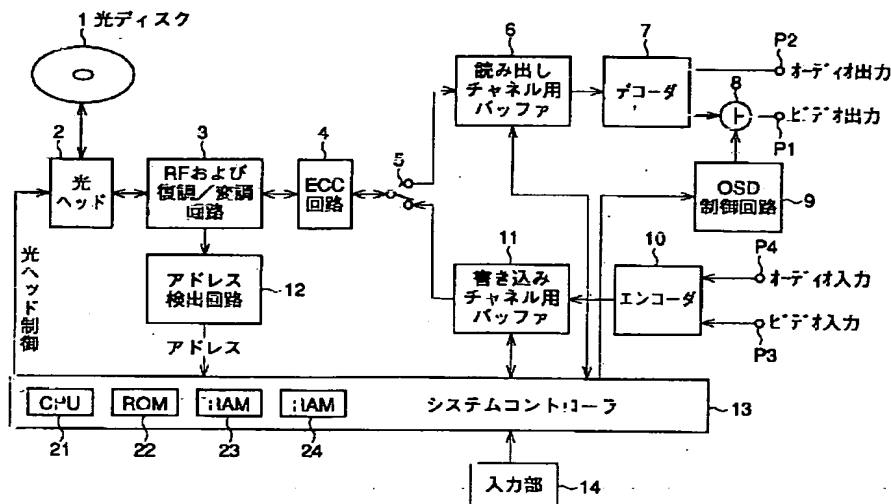
----- VOLUME.TOC
|
|-----ALBUM ST:1
|
|-----PROGRAM
|
|-----PROGRAM_001. PGI
|
|-----TITLE
|
|-----TITLE_001. VDR
|
|-----TITLE_002. VDR
|
|-----TITLE_003. VDR
|
|-----CHUNKGROUP
|
|-----CHUNKGROUP_001. CGIT
|
|-----CHUNKGROUP_002. CGIT
|
|-----CHUNK
|
|-----CHUNK_0001. ABST
|
|-----CHUNK_0011. ABST
|
|-----CHUNK_0012. ABST
|
|-----MPEGAV
|
|-----STREAMS_001
|
|-----CHUNK_0001. MPEG2
|
|-----STREAMS_002
|
|-----CHUNK_0011. MPEG2
|
|-----CHUNK_0012. MPEG2

```

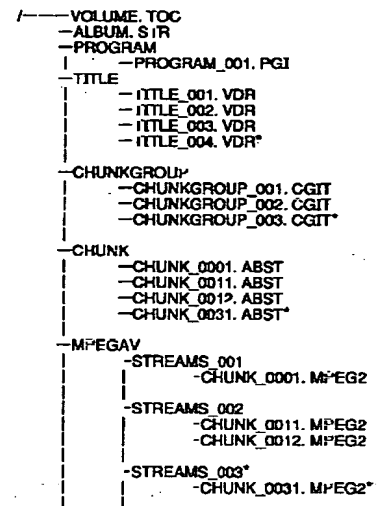
【図27】



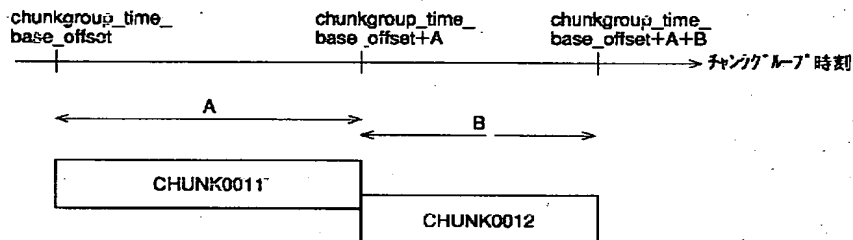
【図25】



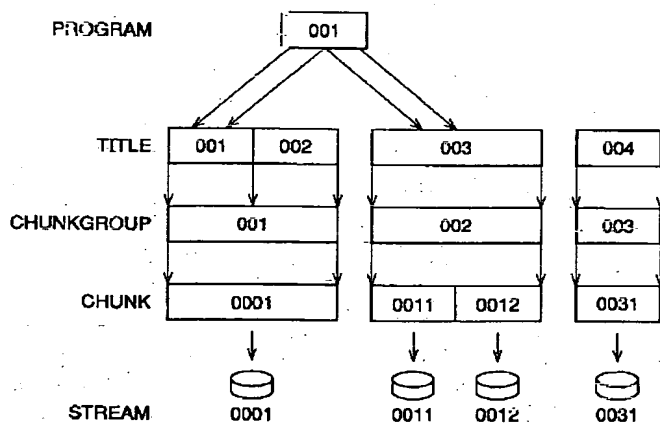
【図30】



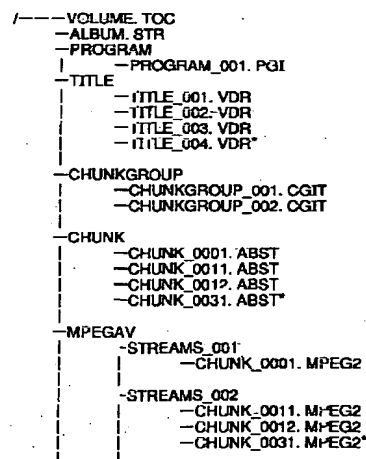
【図28】



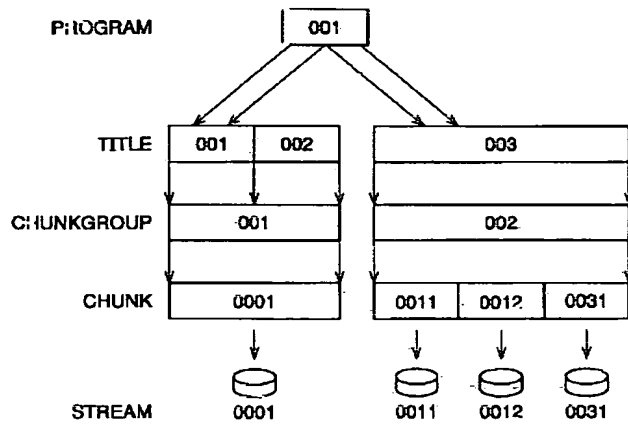
【図31】



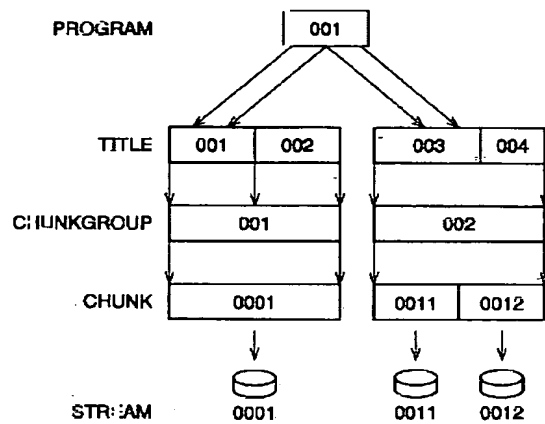
【図33】



【図34】



【図35】



【図36】

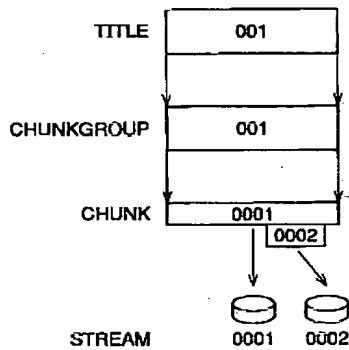
field name	value
file_type_id	"TITLE_INFO_00000"

【図37】

mark_type	Meaning
0000b	index type 1
0001b	index type 2
0010b .. 0111b	index type 3 .. index type 8
1000b	skip IN
1001b	skip OUT
1010b	jump to title end or next title
1011b	scene change
1100b	audio mute
1101b	audio peak
1110b	picture still
1111b	reserved

index : direct entry point in the title  
 skip IN : start point of title skip  
 skip OUT : end point of title skip

【図38】



【図40】

field name	value
file_type_id	"CHUNKGROUP_0000"

【図41】

chunk_sync_play_flag	Meaning
0b	play only one chunk at the same time
1b	need to play chunks simultaneously

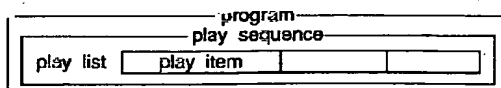
【図43】

field name	value
file_type_id	"STREAM_INFO_0000"

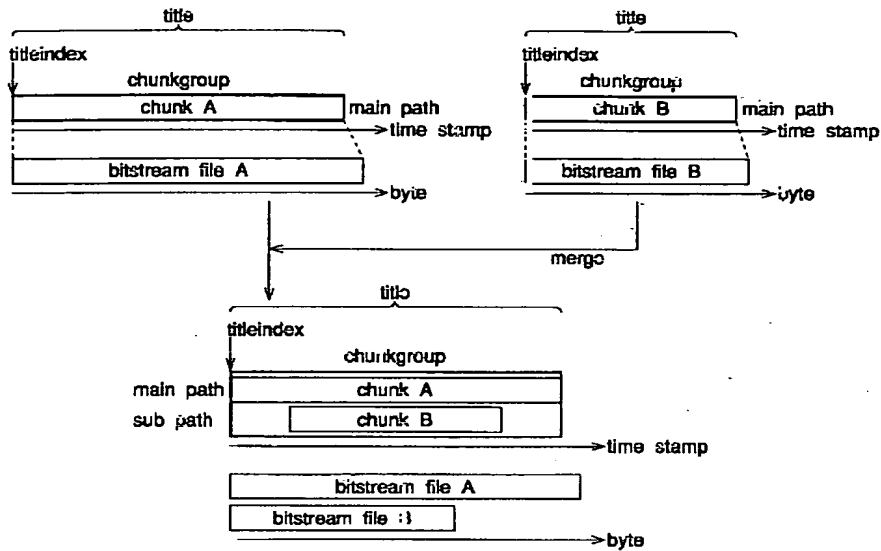
【図46】

field name	value
file_type_id	"PROGRAM_IN:O_000"

【図60】



【図39】



【図42】

original_time_count_type	Meaning
0000	MPEG2 System time stamp(90kHz)
0001	SMPTE timecode
0010	field(525/60)(59.94Hz)
0011	field(625/50)
0100	frame(525/60)(29.97Hz)
0101	frame(625/50)
0110	drop frame time code(525/60)
0111	non drop frame time code(525/60)
1000	60Hz
1001	50Hz
1010	24Hz
1011	second
1100 .. 1111	reserved

【図45】

slot_unit_type	Meaning
0000b	'time_stamp' : time stamp value
0001b	'GOP' : one GOP(Group of pictures)
0010b	'audio_frame' : one audio frame
0011b .. 1111b	reserved

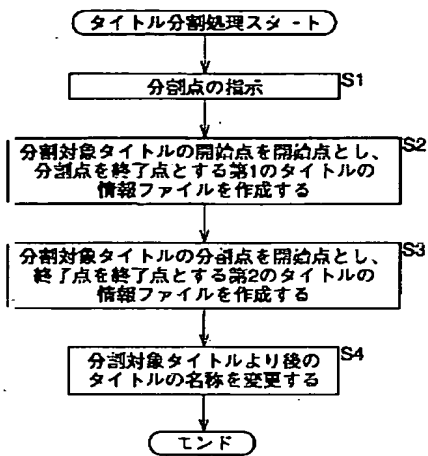
【図44】

Info_type	Meaning
0000	MP::G2_System_PS
0001	MP::G2_System_TS
0010	MP::G2_System_PES
0011	video_elementary_stream
0100	elementary_stream_except_video
0101	MPEG1_System_stream
0110	reserved
0111	reserved
1000	Consumer_DVC
1001 .. 1111	reserved

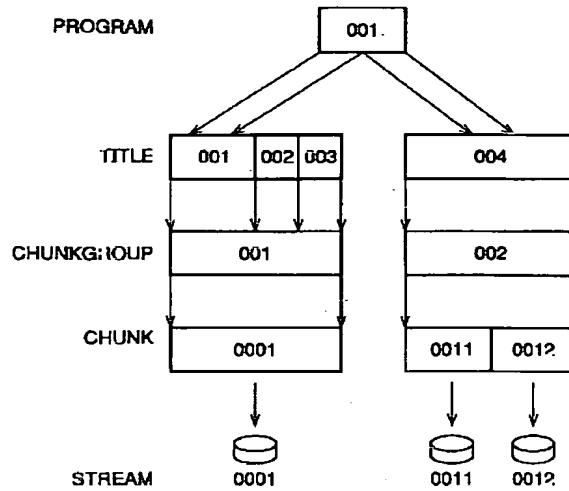
【図47】

program_status	Meaning
0000b	none
0001b	original
0010b	copy
0011b	preview
0100b	rehearsal
0101b	temporary
0110b	complete
0111b	broken
1000b	editing
1001b	backup
1010b .. 1111b	reserved

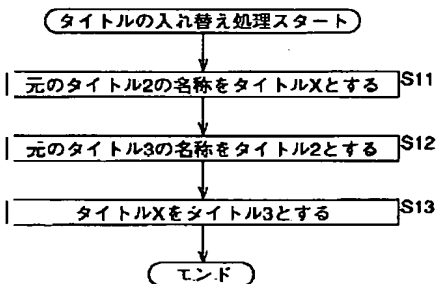
【図48】



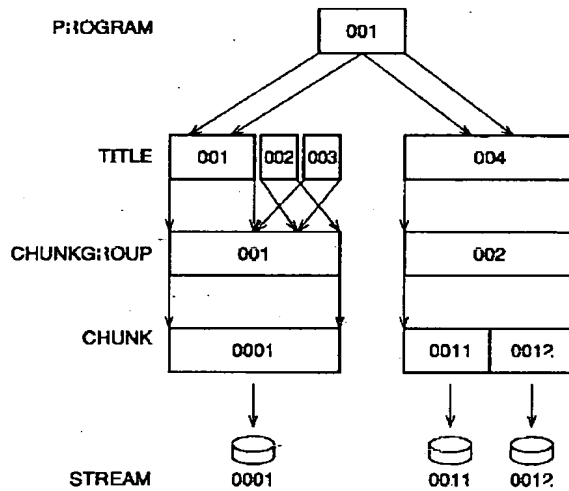
【図49】



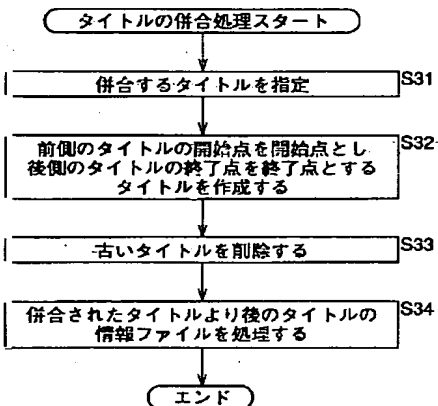
【図50】



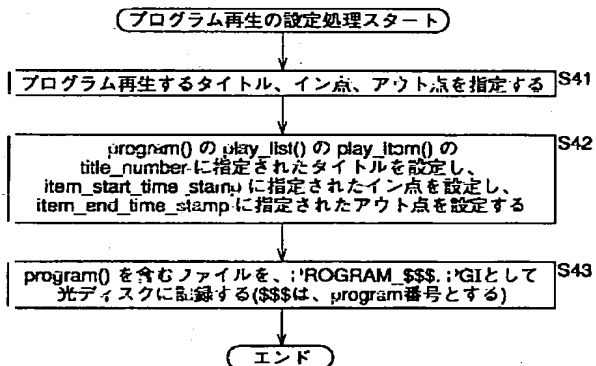
【図51】



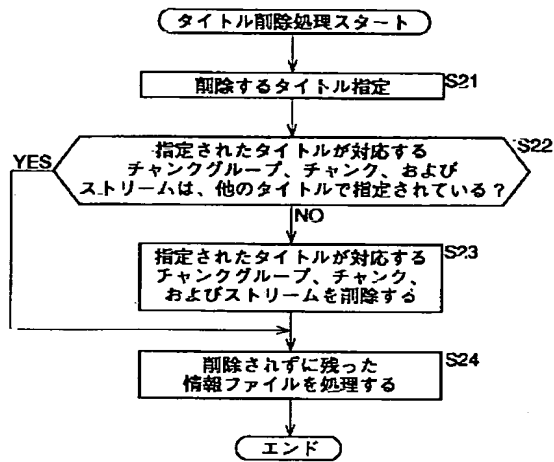
【図54】



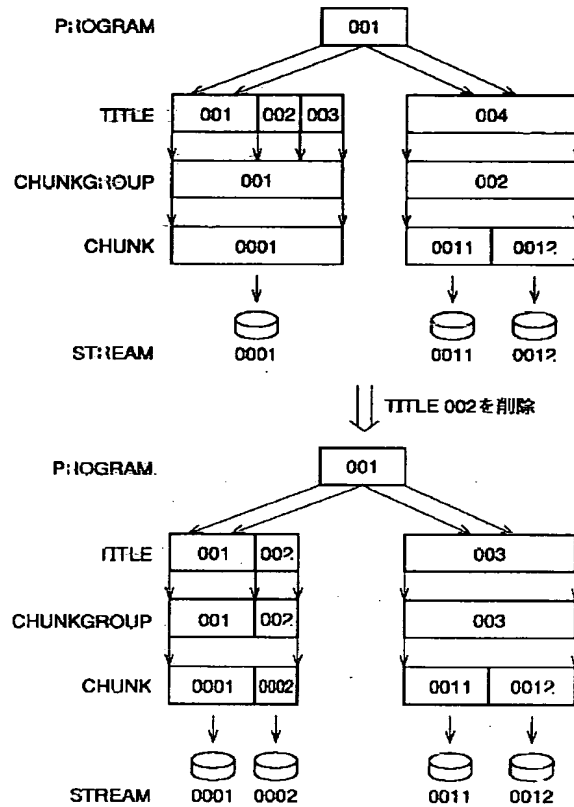
【図55】



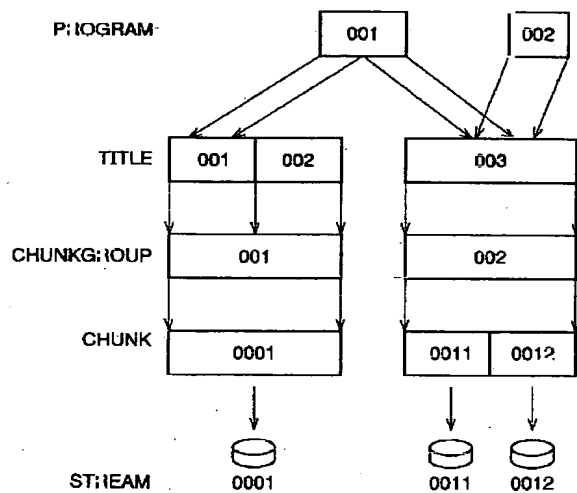
【図52】



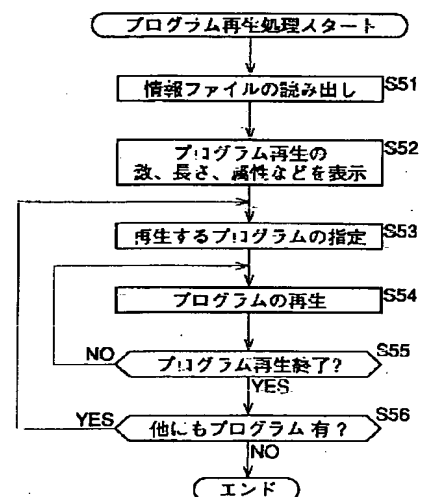
【図53】



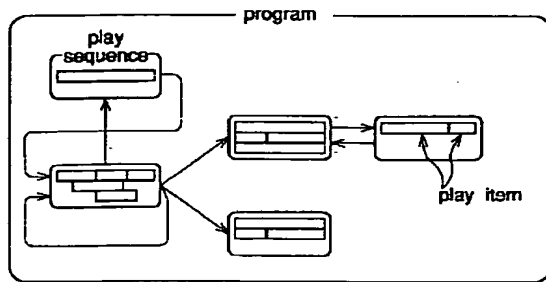
【図56】



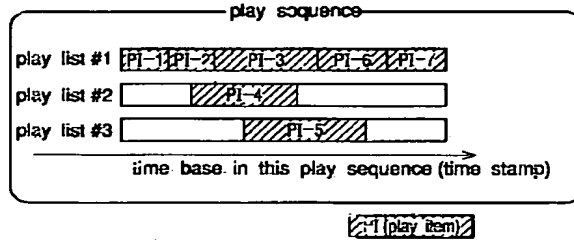
【図57】



【図58】



【図59】



## 【手続補正書】

【提出日】平成10年10月23日(1998. 10. 23)

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0041

【補正方法】変更

## 【補正内容】

【0041】図1のCHUNK\_%%.MPEG2のファイルは、ストリームファイルである。このファイルはMPEGのビットストリームを格納しており、その他のファイルが情報のみを記録しているのと異なっている。

## 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0177

【補正方法】変更

## 【補正内容】

【0177】すなわち、最初に、CPU21は、VOLUME.TO CとALBUM.STRを読み出し、次に、ファイルシステムに対し、ディレクトリ"program"以下に、".PGI"の拡張子を有するファイルがいくつあるかを問い合わせる。この拡張子を持つファイルは、プログラム再生の情報を持つファイルであり、そのファイルの数がprogram再生の数となる。図27の例の場合、この数は1となり、図56の例の場合、この数は2となる。

## フロントページの続き

(31)優先権主張番号 特願平10-46858

(32)優先日 平成10年2月27日(1998. 2. 27)

(33)優先権主張国 日本(JP)

(31)優先権主張番号 特願平10-120391

(32)優先日 平成10年4月30日(1998. 4. 30)

(33)優先権主張国 日本(JP)

(31)優先権主張番号 特願平10-120389

(32)優先日 平成10年4月30日(1998. 4. 30)

(33)優先権主張国 日本(JP)